



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH  
Escola d'Enginyeria de Telecomunicació  
i Aeroespacial de Castelldefels



Master's degree in Applications  
and Technologies for  
Unmanned Aircraft Systems

# MASTER THESIS

**TITLE:** Drone Based Control of Pine Processionary Moth Outbreaks in Mediterranean Woodlands.

**MASTER DEGREE:** Master's degree in Applications and Technologies for Unmanned Aircraft Systems (Drones) (MED)

**AUTHOR:** Fabian Christopher Sasse

**ADVISOR:** Dr. David Remondo; Vasili Zhurman

**DATE:** September 7, 2018



**TITLE:** Drone Based Control of Pine Processionary Moth Outbreaks in Mediterranean Woodlands.

**MASTER DEGREE:** Master's degree in Applications and Technologies for Unmanned Aircraft Systems (Drones) (MED)

**AUTHOR:** Fabian Christopher Sasse

**ADVISOR:** Dr. David Remondo; Vasili Zhurman

**DATE:** September 7, 2018

### **Abstract**

Unmanned aerial vehicles (UAVs) are a rapidly evolving technology that has great potential in various applications, such as surveying, precision agriculture or search and rescue. The autonomy of drones and their embedded sensors allow to gather data and to perform mechanical tasks in areas that are only accessible from the air. Drones allow for a quick and inexpensive operation when compared to traditional methods used for the applications mentioned above.

The objective of this thesis was to research, if a drone based system for the control of Pine Processionary Moth (PPM) outbreaks in the Mediterranean woodlands is a feasible concept. The insect builds silken nests in the tip of pine tree branches to survive the cold temperatures during winter months. The concept developed for the body of this thesis uses a drone to detect said nests and to inject pesticides into its center to eradicate its inhabitants.

The system's feasibility was studied by implementing the system as a pilot project in pine woodland stands in the area of Catalunya, Spain. Literature related to the use of drones in similar applications was reviewed, from which requirements for a successful system design were derived. The system was developed with attention to hardware and software performance according to these requirements. The completed system was evaluated by comparing the system's performance against traditional pest control methods and by assessing how well it met the requirements. From the results of this evaluation, the feasibility of the system was assessed and recommendations for further improvements were given. The research confirmed that a drone based control system for PPM outbreaks is a feasible and practically working concept. The system can treat infected pine trees with a high level of accuracy and sustainability while maintaining low costs for operation; It is therefore outperforming, and capable of replacing, more traditional pest control methods. Current limitations of this concept are battery life, autonomy and lack of financial resources. Improvements in these aspects could greatly enhance the work-flow and the overall system.

---

## **Acknowledgements**

I would like to express my sincere gratitude to my thesis advisor Prof. Dr. David Remondo for the continuous support over the process of writing this thesis and for his his patience, motivation, and knowledge.

I would like to thank Prof. Dr. Esther Salamí San Juan and Prof. Dr. Pablo Royo Chic for their encouragement and helpful advice on technical challenges.

My sincere thanks also goes to Toni Caballero and Vasili Zhurman, who invited me to their team as intern. Without their support, mentoring and access to their test facilities I could have never conducted this research.

I especially appreciate the support by Prof. Dr. Miguel Valero-Gracia, who has always been there for his students. His motivation and excitement for this course was contagious and I wish him all the best for next year's program. Thank you for making this year such a great experience.

Last but not the least, I would like to thank my girlfriend, family and friends for supporting me spiritually throughout this year.

## List of Figures

1.1	Silken nest with caterpillars. . . . .	1
1.2	Pine woodland stands in Spain, representative for main distribution range of PPMs [2]. . . . .	5
1.3	Number of pine stands and woodland area in Andalusia [4]. . . . .	7
1.4	Defoliation degree of treated (solid) and untreated (dotted) pine trees [4]. . . . .	7
1.5	Illustration of fixed-wing and multi-rotor UAV. . . . .	9
2.1	Mission work-flow diagram. . . . .	16
2.2	Prototype models. . . . .	17
2.3	CAD model of FSD with part index according to Table 2.1. . . . .	18
2.4	CAD model of DJI Flamewheel F450 frame part. . . . .	19
2.5	Top view of FSD. . . . .	20
2.6	T-Motor F40 Pro-II thrust table [22]. . . . .	21
2.7	Application to determine HSV value range. . . . .	28
2.8	Image with colour based object detection. . . . .	30
2.9	Nest detection from video recording. . . . .	32
2.10	Test set-up for video transmission to ground station. . . . .	34
2.11	Pine tree in Serra de Montgrony with 15+ nests. . . . .	35
3.1	Correlation of tip displacement and pitch angle. . . . .	41
A.1	Code to determine HSV range . . . . .	52
A.2	Code for detection from live camera stream. . . . .	53
A.3	Three views of FSD CAD-model. . . . .	54

## List of Tables

1.1	List of requirements . . . . .	12
2.1	Prototype comparsion . . . . .	18
2.2	Maximum discharge current of selected batteries. . . . .	23
2.3	Battery-life test data for <i>Octacopter v2.1</i> . . . . .	36

## List of Source Codes

1	Creating Mouseclick-events. . . . .	25
2	Performing color-space conversion. . . . .	25
3	Creating a masked image with desired color spectrum. . . . .	26
4	Creating trackbar application. . . . .	27
5	Applying morphological transformations. . . . .	29
6	Finding contours within the image. . . . .	30
7	Drawing contours around detected object. . . . .	30
8	Read video recording into detection algorithm. . . . .	31
9	Apply filter to remove false-positive objects. . . . .	31

## Glossary

**BGR** Blue, Green, Red. 25

**BIC** Business Incubation Center. 14

**ESA** European Space Agency. 14

**FAA** Federal Aviation Administration. 8, 10–13, 35, 45

**FPV** First-Person-View. 43

**FSD** FitoStinger Drone. 2, 8–10, 12, 13, 22, 33, 35, 36, 45, 46

**GNSS** Global Navigation Satellite System. 40

**GPS** Global Positioning System. 40

**HSV** Hue, Saturation, Value. 25

**LiPo** Lithium Polymer. 22, 23

**LOS** Line-of-Sight. 11

**NDVI** Normalized difference vegetation index. 9

**NIR** Near-infrared. 9

**OEW** Operational Empty-Weight. 18

**OS** Operating System. 43

**PPM** Pine Prosessionary Moth. 1, 2, 4–12, 14, 35, 44, 45

**RPI** RaspberryPi. 33, 34, 37, 38

**SPOF** Single Point of Failure. 12

**TRL** Technology Readiness Level. 12

**UAV** Unmanned Aerial Vehicle. 4, 8–11, 44



# Contents

List of figures . . . . .	
List of tables . . . . .	
List of source codes . . . . .	
Glossary . . . . .	
<b>1 Introduction</b>	<b>1</b>
1.1 Research Methodology . . . . .	2
1.1.1 Research method . . . . .	2
1.1.2 Literature review method . . . . .	4
1.2 Literature review . . . . .	4
1.2.1 The Pine Processionary Moth . . . . .	5
1.2.2 Current pest control techniques . . . . .	6
1.2.3 Key drivers for alternative methods . . . . .	8
1.2.4 Off-the-shelf technology . . . . .	8
1.2.5 Benefits of drones in pest control . . . . .	10
1.2.6 Limitations of drones in pest control . . . . .	10
1.2.7 Market Study . . . . .	11
1.2.8 Derivation of requirements . . . . .	12
<b>2 Design</b>	<b>14</b>
2.1 Previous work . . . . .	14
2.2 Mission design . . . . .	15
2.3 Hardware design . . . . .	17
2.3.1 Prototype comparison . . . . .	18
2.3.2 Hardware component selection . . . . .	19
2.4 Software design . . . . .	24
2.4.1 Methodology . . . . .	24
2.4.2 Color based object detection . . . . .	24
2.5 Testing the system . . . . .	35
2.5.1 Testing the hardware . . . . .	36
2.5.2 Testing the software . . . . .	37
<b>3 Evaluation</b>	<b>39</b>
3.1 Hardware . . . . .	39
3.2 Software . . . . .	41
<b>4 Discussion</b>	<b>44</b>
4.1 Sustainability . . . . .	44
4.2 Ethics . . . . .	45
4.3 Future Research . . . . .	46
<b>5 Conclusion</b>	<b>47</b>

**Appendices**

<b>Appendix A</b>	<b>52</b>
-------------------	-----------

# Chapter 1. Introduction

Insect pests are a widely spread threat to forestry world wide. The pine processionary (*Thaumetopoea pityocampa*) is part of the moth family, that is mostly found on pine trees and cedars in Central Asia, North Africa and Southern Europe. The PPM is one of the most destructive pests in Mediterranean countries [1]. During the caterpillar phase, they are an imminent threat to host trees by feeding on their needles. Also, direct contact with their hair causes harmful reactions to humans and other mammals. For this reason, the caterpillar has basically no natural predators. To survive cold temperatures during the winter months, the caterpillars build silken nests in the tips of tree branches, see Figure 1.1. This strategy in combination with raising temperatures and the lack of natural predators has allowed this species to reproduce and propagate without restriction.



Figure 1.1: Silken nest with caterpillars.

For the scope of this study, the author analysed the performance of aerial pesticide spraying over pine forest areas, to restore a balanced population of the PPM in north-east Spain. Instead of spraying pesticides over the entire forest area and thereby threatening plants and other animals, a fleet of drones will autonomously detect and directly insert pesticides into the nests. Starting with a manually piloted and rather complex drone system, this document is a step-by-step progress report that highlights the advances made with the intelligent drone for plague control in Spanish forestry.

The document is structured as follows. The study begins in Section 1.1 with a definition of the research problem and specification of the research questions. With the research methodology defined, Section 1.2 highlights the results of the structured literature review that was conducted to collect existing information related to drones in similar applications. It also presents the latest update on the PPM population in north-east Spain. Furthermore, it highlights current mitigation methods and entails a preliminary comparison between these and the newly proposed method. Based

on this information, the section ends with the derivation of requirements for the Fi-toStinger Drone (FSD). The system design is elaborated upon in Chapter 2. This entails the mission, as well as the hardware and software design and ends with the testing of the prototype system. The results of these tests are analyzed and evaluated in Chapter 3. This is followed by a discussion in Chapter 4 on sustainability, the ethical questions that arise when conducting a wildlife control operation and recommendations for future work on this concept. The document ends in Chapter 5 with the conclusion. Additionally, the Appendix includes 3D drawings of the prototype as well as details on the algorithm used for the detection of the nests.

## **1.1 Research Methodology**

This research was conducted to assess the feasibility of a drone based system for the control of the PPM pest in pine woodland stands around the globe. Further elaboration on this objective was necessary to derive the appropriate research questions. As mentioned in the introduction, the system consists of a drone capable of detecting PPM population and eradicating its stands by directly inserting lethal chemicals into their nests. The research problem was initiated by the TSA Center, a company that designs technical solutions for current problems in the area of Catalonia, Spain. However, aggravating conditions world wide set the perspective from which the problem was approached. The system was finally evaluated by assessing the compliance with its design requirements and by comparing it to traditional prevention methods. From this, the following five research questions were derived:

1. What are traditional methods used for insect pest control?
2. What are the limitations of the traditional methods?
3. How to design an alternative, drone based method?
4. What are the benefits of a drone based system in the control of insect pests?
5. What are its limitations?

Answering the first question established a foundation of information on the origin and current status of the insect pest, and presents previous attempts to decrease its propagation. Answering the second question highlighted the limitations of these methods and helped to identify the points of failure. From this analysis followed the derivation of requirements for the alternative system. The design and testing of the prototype finally uncovered the benefits and limitations of the suggested approach.

### **1.1.1 Research method**

The research was performed as an iterative process of multiple stages. Since this system is the precursor of its kind, this document describes the complete development process. The following stages were repeatedly performed over the course of the research:

**Stage 1: Derivation of requirements**

The objective of this stage was to develop a list of requirements for the system. The requirements were derived from the results of the literature review and the requirements given by the company. The validity of the requirements was an important factor for the structured design of the mission and the system, and should therefore be complete and consistent before the start of the development process. However, the experience gained over the course of the project gave rise to new requirements or highlighted flaws in existing ones. Therefore, this list was a living document that had to be checked constantly for its validity and completeness.

**Stage 2: Design of the mission structure**

This stage determined the work-flow of the system, see Figure 2.1. This included all the all the activities performed during the mission. The components and work-flow of the mission were derived from the requirements of the previous stage. It was assessed by checking the compliance of mission components and work-flow with the list of requirements.

**Stage 3: Design of the drone platform and payload**

The objective of this stage was to choose a drone platform and design its payload in compliance with all previously derived requirements. The drone had to be capable of performing all the activities and be able to follow all components of the work-flow-diagram. The evaluation of this stage was performed in stage 6, after testing the system.

**Stage 4: Design of the software**

The objective of this stage was to develop a software algorithm that meets all the previously derived requirements. The algorithm had to be capable of performing all the activities and be able to follow all components of the work-flow-diagram. The evaluation of this stage was performed in stage 6, after testing the system.

**Stage 5: Testing the system**

At this stage a system prototype was built to test its performance in all the mission features according to the work-flow-diagram and system requirements. To do so, the individual hardware and software components developed in the previous two stages were used in an integrated system prototype.

**Stage 6: Evaluating the system**

The objective of this stage was to evaluate the test results from the previous stage. The completion of the stage was evaluated by confirming that the system is capable of performing all mission features to a satisfactory level. If the the system did

not perform correctly, the points of failure were analyzed, and the iteration process restarted.

### **1.1.2 Literature review method**

The literature review started with a broad search that was gradually narrowed down to get the most relevant results. The goal was to better understand the nature of the threat and to avoid common mistakes in the development of a drone based system for pest control. The study only included research articles, conference papers and books, gathered from platforms such as Google.scholar and Researchgate. The bulk of information was eventually reduced to a reasonable amount of literature for the scope of this thesis.

A specific difficulty that was often encountered is worth mentioning here. The search mainly lead to papers on drone systems using conventional aerial spraying methods, as it can be commonly found in agriculture. Articles on drone systems with a retracted arm for surgical injection of pesticides were not found. The decision to include a research paper into the literature review was done by recognizing whether the title of the paper is relevant to one of the following questions:

- Does the paper focus on mitigation techniques for the PPM pest?
- Does the paper focus on the application of pesticides using UAVs?
- Does the paper focus on the detection of objects using aerial imagery?

The list of literature used for this study can be found at in the list of references at end this document.

## **1.2 Literature review**

This section presents the results of the literature review described in Subsection 1.1.2. The goal of the literature review was to get a deeper understanding of the threat caused by the PPM pest, to get a better insight into traditionally prevention techniques, and to find inspiration in previous work on drone based systems for pest control. The outcome was used to develop a list of mission and system requirements. Research on the problem and previous work was an important factor in understanding how the system should be developed and what aspects should be taken into consideration for its implementation.

The first part of the literature review focuses on the fundamental facts about the problems that accompany the over-population of PPMs in pine woodland stands. This gives the reader a brief insight into the life cycle of the PPM, reasons for its rapid territorial expansion and historical data on its population numbers in recent years. The following section reveals traditional methods to cope with the threat. A

particular study, which will be mentioned in detail, compares the decrease in defoliation levels of pines trees between treated and untreated trees, and thereby discusses the effectiveness of these methods. The section ends with the study of existing, drone based, pest control methods. It addresses the difference between traditional drone based methods and the approach suggested in this thesis. It also points out the limitations and benefits of using drone based systems in the attempt to control the outbreak of PPM populations.

### 1.2.1 The Pine Processionary Moth

The PPM is one of the most known and well studied species of processionaries. The insect is mainly located in warmer regions of Southern Europe, the Near East and North Africa [4]. It is active during the colder seasons of the year, late Autumn to early Spring. During the warmer summer months it buries itself in the ground, until finally emerging from the soil again in late August to to find pine trees to feed on and lay its eggs. During the PPMs population surge, it is capable of defoliating vast areas of pine stands. A PPM colony in regions near Barcelona, Spain, was observed with infrared activity monitoring, showing the behaviour of the insect by night. The caterpillars leave the nest right after sunset to feed on the pine needles, thereby often exposed to sub-zero temperatures. Around March the caterpillar will have grown to its full size, at which point the colony moves out in long, head-to-tail processions in search for population sites in the soil. In this form the caterpillar is often encountered by hikers and dog-walkers trying to enjoy the first warm days of spring. Many of the accidents happen during that period.

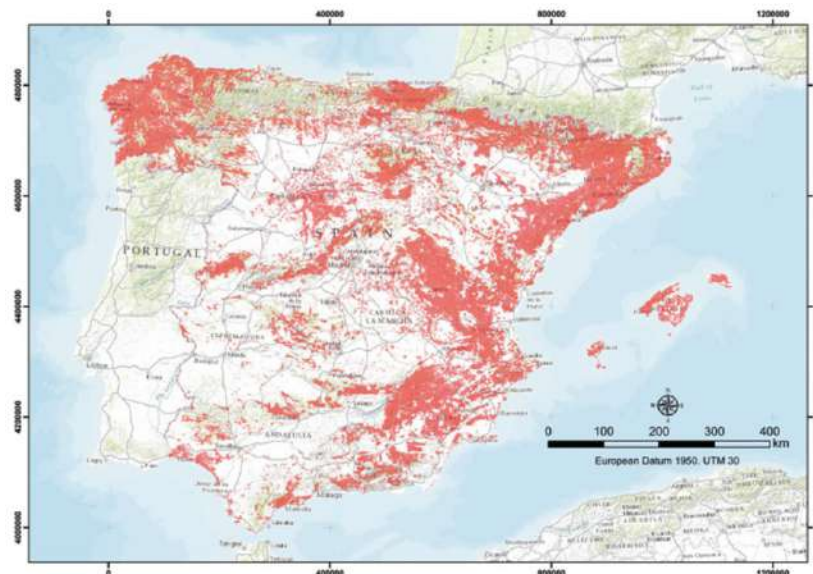


Figure 1.2: Pine woodland stands in Spain, representative for main distribution range of PPMs [2].

Over the course of the last decades, the spread of the PPM pest has increased and their population has expanded further to the north of southern Europe [2]. It has reached the higher altitude pine woodland areas of the Pyrenees, as it can be seen in Figure 1.2. High incident rates between the PPM, humans and other animals have been reported in areas that have previously been unaffected by the pest. This rapid increase calls for more sophisticated and efficient control techniques and is of major importance for the Mediterranean forestry and its inhabitants.

### 1.2.2 Current pest control techniques

Currently used prevention measures include the manual cutting and burning of nests, pheromone traps, mating disruption systems and most commonly used, the distribution of chemical pesticides by aerial spraying using fixed wing aircraft [4]. Several studies were conducted on the effects of the pesticides, with respect to the containment of the PPM pest as well as the impact on the environment. The results were interpreted as the method being successful in reducing further spreading of the pest while causing no harm to other wildlife. However, the studies were performed at a stage where the population was expected to reduce as a result of natural balance. Once a species like the PPM exceeds a certain threshold, the lack of host trees, diseases or natural predators cause the population to naturally reduce until it is once again in balance with the ecosystem. This gives rise to the question whether the decrease in PPM population was actually a consequence of the mitigation methods, or simply a result of the moth's natural population cycle.

The study was conducted in the region of Andalusia, in southern Spain. It covers around 87,300 km<sup>2</sup>. It consists of different types of vegetation and landscape. The performance of the application of traditional methods was evaluated by comparing the defoliation process of treated and untreated trees over the course of four years. The hypothesis of the study was, that if the outbreak of the pest is controlled by natural means, there would be no difference in defoliation levels between treated and untreated trees.

Figure 1.3 shows the total amount of pine tree stands in terms of single stands (N) and total area (km<sup>2</sup>). It gives the percentage of highly defoliated trees as well as the number of stands exposed to treatment. The figure shows that the number of highly defoliated pine stands increased constantly between 2002 and 2004 from 9.5 % to 11.7%, respectively. A similar pattern can be observed for the forest area, increasing from 10.3% to 12.1%. Realizing that during that period the treated forest area decreased, it could be concluded that there is a connection between the defoliation and treatment. However, in 2005, the treated forest area slightly increased again from 223 km<sup>2</sup> in the previous year to 250 km<sup>2</sup>. On the contrary, the area with highly defoliated trees decreased rapidly from 12.1% to 6.7%, which equals an improvement of 50% within one year. If this were a reaction to the chemicals, similar effects should be observable in earlier years when the area of treatment was adjusted. Therefore, the authors attributed the recession of the PPM to exhaustion of



resources and an increase of natural enemies (predators, parasites) [4].

	2002		2003	
	N stands	km <sup>2</sup>	N stands	km <sup>2</sup>
Total pine forest stands	3350	6192	3597	6566
Degree of infestation $\geq 3$	318 (9.5%)	639 (10.3%)	398 (11.1%)	715 (10.9%)
Total treated	170	312	138	250
Insecticide spraying	149	280	135	245
Other treatments	21	32	3	5
	2004		2005	
	N stands	km <sup>2</sup>	N stands	km <sup>2</sup>
Total pine forest stands	3639	6724	3656	6702
Degree of infestation $\geq 3$	427 (11.7%)	813 (12.1%)	218 (6.0%)	453 (6.7%)
Total treated	118	223	111	250
Insecticide spraying	118	223	110	249
Other treatments	0	0	1	1

Figure 1.3: Number of pine stands and woodland area in Andalusia [4].

Figure 1.4 gives another indication that traditional methods are not effective to control the outbreak of the PPM. It shows the mean and standard deviation of the degree of defoliation in sprayed (solid line) and unsprayed (dotted line) trees. The study was conducted on four different types of pines (*Pinus nigra*, *Pinus halepensis*, *Pinus pinaster*, *Pinus pinea*). It can be seen that both test groups experience a similar decrease in defoliation, implying that the treatment has no direct impact.

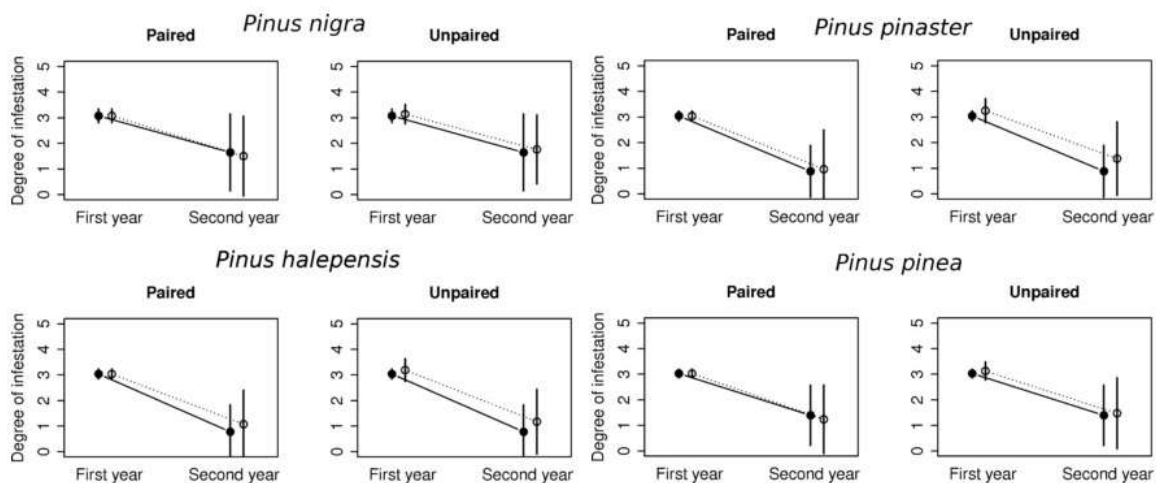


Figure 1.4: Defoliation degree of treated (solid) and untreated (dotted) pine trees [4].

### 1.2.3 Key drivers for alternative methods

According to [4], between € 1.0 and € 1.5 million are spent annually on aerial spraying to control PPM outbreaks in Andalusia. In search for an answer to why traditional methods are unsuccessful, the authors point towards the timing of the treatment. In general, aerial spraying is conducted in the winter following strong, heavy or massive defoliation events. At this stage treatments are unlikely to limit growth losses or prevent further damage to trees by other organisms. One of the main reasons why traditional methods are unsuccessful, is the fact that during daytime the PPM is lingering inside its nest to hide from predators and to increase its body temperature in order to survive the feeding events during the cold hours of the night. If the pesticides are then distributed over vast forest areas by aerial spraying, the PPM inside its nest does not get in contact with the chemical. Therefore, a method for direct insertion of pesticides into the center of the nest would greatly enhance the effectiveness of pesticide spraying. A control of the outbreak is especially crucial in regions close to populated areas. However, conventional aerial spraying is unpractical, due to regulations for low-altitude overflies of populated areas as well as the risk of wind carrying the pesticide in areas with human presence. Alternatively, pheromone traps, hand removal, or application of high insecticide concentrations using truck or back-pack methods heavily increase the risk for the treaters.

### 1.2.4 Off-the-shelf technology

As previously mentioned, no example for drones using precision injection of pesticides in forestry was found. The closest relatives to such drones are those used for precision agriculture. The different software and hardware components necessary for this application are highly relevant for the design of the FSD. Therefore, this section highlights the most successful drones in precision agriculture in 2018.

#### 1.2.4.1 Product availability

Drones are becoming increasingly accessible for commercial use. There is a broad range of ready-to-fly camera and race drones on the market for amateur drone hobbyists. Agricultural drones however do not fall into this category. The use of such drone falls strictly under the regulations governed by the FAA. Therefore, every agricultural drone operator is required to carry a Remote Pilot Certificate, every single survey or spraying mission needs to be planned according to the rules [13].

There are several ready-to-fly UAVs for precision agriculture, both fixed-wing and multi-rotor configuration. This section will start elaborating on fixed-wing systems, which are especially useful for surveying of large areas, up to ten times the acreage of typical multi-rotor drones [5]. Their disadvantage comes with the image quality deterioration due to the speed of the vehicle. Although this type of UAV cannot be used for the injection of pesticides, it is a great tool for the mapping of the area and the scouting of PPM population hot-spots.

Figure 1.5: Illustration of fixed-wing and multi-rotor UAV.



(a) Fixed-wing UAV for surveying.



(b) Multi-rotor UAV for aerial spraying.

### Fixed-wing UAVs

A popular system is the *eBee SQ* (Figure 1.5a), a fixed-wing UAV by *Parrot*, one of the worlds leading drone companies [5]. It is an off-the-shelf system, starting at 12.000 USD. So far, this series has logged over 300.000 missions world-wide. Its on-board multi-spectral and visual sensors allow the creation of vegetation index maps, and are used to gather temperature information and plant counts. One of its limitations is the incapability of creating topography and precision 3D mapping.

Another, probably the most advanced and intelligent fixed-wing system available for precision agriculture, is the PrecisionHawk Lancaster 5, starting at 25.000 USD [5]. Equipped with the widest range of sensors and artificial intelligence, the system can not only measure vegetation indices, humidity, temperature, air pressure and incident light, it also autonomously adapts to changing weather conditions and payloads. It can create highly accurate and geo-referenced 2D and 3D maps of its environment, including plant counts and plant height. This is a valuable tool for the control of PPM population, as it can be used in combination with the FSD to achieve its final goal of autonomously detecting and eradicating PPM nests.

### Multi-rotor UAVs

A suitable solution for close range and detailed investigation of vegetation are multi-rotor UAVs. Their low speed and stability allow higher accuracy and better image resolution, for the cost of area coverage.

A widely popular and affordable multi-rotor agricultural drone is the DJI Phantom 4 PRO with TrueNDVI camera updated by Sentra [5]. Starting at roughly 2.000 USD, this upgraded off-the-shelf UAV can generate Normalized difference vegetation indices (NDVI) using a 1.2MP Near-infrared (NIR) global shutter camera, while the build in Phantom camera still produces color images for observations and piloting of the drone. It comes with a free license for the AgVault image processing software.

This tool allows the operator to quickly define a mission area for the drone to survey. Upon completion, the data, in this case the indication of trees with high degrees of defoliation, can be accessed immediately and compared to data history to see possible effects of previous treatment.

The final drone model to be mentioned for this research is the *Agras Mg-1* by DJI (Figure 1.5b) [7]. Costing roughly 10.000 USD on Amazon, this drone's powerful propulsion system allows the drone to carry up to 10kg of liquid payload, while covering an area of 4.000-6.000 m<sup>2</sup> in just 10 minutes. It has an intelligent spraying system integrated to its motors that functions as conventional aerial spraying methods do, by distributing pesticides evenly from above. The drone has many additional highlights, such as its intelligent memory and foldable design, but one outstanding characteristic of the drone is its high hover accuracy. As it will be discussed later in Chapter 3, a stable hovering ability is crucial for the task laid out for this project.

### 1.2.5 Benefits of drones in pest control

Unmanned Aerial Vehicles (UAVs) equipped with remote image sensing cameras have high potential in pest control applications. UAVs can access woodlands and remnant bush sites that previously have been inaccessible to traditional pest control techniques due to their size and mobility. Drones can not only detect the presence of PPM, but also record the spectral signature of the vegetation to survey the expansion of the PPM population [3]. They are capable of approaching the nests to close vicinity, allowing them to efficiently inject pesticides into their center. From an economical perspective, UAVs are considerably cheaper to purchase and operate than conventional aircraft used in aerial pest control operations [8]. Furthermore, if equipped with the right vision sensors, drones are capable of working at night, which could result in a faster elimination of the threat. However, current FAA regulations force the operator to apply for a *Part 107* authorization before performing night operations [9]. From the ecological perspective, traditional spraying techniques create runoffs, that pollute our water sources, kill animals not targeted by the operation and even pollute the air by its lack of accuracy; It is an overall dangerous and resource wasting practice [10]. This research aims to confirm that the FSD on the contrary applies pesticides at a high level of accuracy, and is thereby eliminating collateral damage.

It can be concluded that a directed and targeted approach using UAVs could be a valuable and novel tool in the pest management toolbox that could significantly reduce pest control costs and cover inaccessible areas not receiving any pest management, while also reducing unwanted impacts to the ecological system.

### 1.2.6 Limitations of drones in pest control

As concluded in the previous section, UAVs could be an effective tool in pest control management. However, being a fairly new technology, drones have several limita-

tions that yet need to be overcome. This section will highlight the most impacting ones.

A general challenge that arises when planning a UAV based mission, is the decision on which technology to use. An rapidly increasing number of drone services make it a difficult task to choose the right system, from both, a technical and economical perspective [11]. Another challenge is the initial cost intensive investment, especially for small, individual farmers. Although the cost of the UAV can be minimized, the installation, configuration and maintenance of the system requires the advanced knowledge of technicians or engineers, which on the long term increases the total cost. Considering the original applications of drones, such as surveying, mapping and inspection, the structure of commercial drones are not designed to carry heavy payloads such as pesticide tanks. Therefore, it is likely that modifications to the structure are necessary to accommodate the new payloads. Being subject to FAA regulations, agricultural drones need to be operated by licensed pilots, indicating another field to allocate resources to. Last but not least, a difficult challenge especially relevant for the scope of this study, is the operation in remote areas. Flying in elevated and canopy rich areas can result into multipath problems as well as loss of direct Line-of-Sight (LOS) between GPS satellites and UAV [12]. This leads to noncompliance with the cm-accuracy requirement for drones working in precision agriculture.

It becomes clear that that the current challenges associated with UAVs such as payload capacity, battery life, sensitivity to environmental conditions, and airspace regulations are allowing only slow integration of UAVs into existing applications. However, these issues are rapidly being resolved with advancements in technology and the relaxation of laws.

### **1.2.7 Market Study**

According to PWC study, currently \$127.2bn value of global market for drone powered business solutions [6]. With infrastructure being the biggest sector with \$45.2bn, the agricultural sector follows close behind with a total of \$32.4bn. As previously mentioned in Subsection 1.2.3, between € 1.0 and € 1.5 million are spent annually on aerial spraying to control PPM outbreaks in Andalusia [4]. This proves the existence of a big market for drones in the agricultural sector, that is expected to grow with improvements in the technology for unmanned aerial systems.

### 1.2.8 Derivation of requirements

Based on the literature study and the restrictions given by the company and stakeholders, the following list of requirements has been derived for the system. The list is separated in the different system segments. The requirement abbreviation is defined as [< system > < segment > < indication > < numbering >].

Table 1.1: List of requirements

Segment	Key requirements
FSD-KEY-SYS-01	The system shall have no Single Point of Failure [SPOF].
FSD-KEY-SYS-02	The system hardware shall have a minimum Technology Readiness Level [TRL] of five.
FSD-KEY-SYS-03	A clear end-of-life strategy shall be included.
FSD-KEY-SYS-04	The system shall be able to perform a smooth injection of pesticides.
FSD-KEY-SYS-05	The system shall be able to carry all payloads.
FSD-KEY-REG-01	The system shall adhere to all FAA regulations.
FSD-KEY-BAT-01	The system shall be able to at least inject pesticides into one nest per battery pack.
FSD-KEY-PL-01	The pesticides shall have a minimal effect on other biological organisms.
Segment	Platform
FSD-PL-TEMP-01	The temperature in the system shall be within component limits.
FSD-PL-PA-01	The platform shall be able to accommodate a visual sensor with transmitter.
FSD-PL-PA-02	The platform shall be able to accommodate a pesticide container.
FSD-PL-PA-03	The platform shall be able to accommodate a GPS module.
FSD-PL-PA-04	The platform shall be able to accommodate a flight controller.
FSD-PL-PA-05	The platform shall be able to accommodate the injection arm.
FSD-PL-PA-06	The platform shall be able to withstand impacts with branches.
Segment	Payloads
FSD-PA-CAM-01	The visual sensor shall be able to transmit coloured images.
FSD-PA-PEST-01	The pesticides shall be lethal to PPMs.
FSD-PA-ARM-01	The injection arm shall withstand the loads during injection.
FSD-PA-ARM-02	The injection arm shall withstand collisions with branches.

<b>Segment</b>	<b>Batteries</b>
FSD-BAT-STR-01	The batteries shall be exchangeable.
<b>Segment</b>	<b>Stability and Navigation</b>
FSD-NAV-MS-01	The systems shall be able to follow pre-planned missions using open-source software.
FSD-NAV-POS-01	The system shall have a hovering accuracy of 5cm.
<b>Segment</b>	<b>Regulations</b>
FSD-REG-FAA-01	The system shall be in compliance with all regulations stated by the FAA.
FSD-REG-PERM-01	The system shall be able to apply for exception permits from the authorities.

## Chapter 2. Design

This chapter elaborates on the full design process of the system. Section 2.1 indicates what stage the project was in when this study was initiated. Section 2.2 highlights the definition of the different mission segments and tasks that the system had to perform based on the requirements derived in Subsection 1.2.8. It is followed in Section 2.3 by a description of the hardware design, which entails a comparison between two prototypes that were tested simultaneously in Subsection 2.3.1. Next, the design of the system's software is described in Section 2.4. It explains the functioning of the algorithm, the methods that were used for the development and it highlights the adjustments and advancements made on the nest detection software over the course of this project.

### 2.1 Previous work

The FitoStinger project was initially founded in 2017, after the plague of the PPM was recorded to be especially harmful in 2016 [14]. The shortage of rains and the increase of temperatures have favored its overpopulation and caused alterations in its natural reproduction cycle. This resulted in an unprecedented number of pine defoliation levels and PPM-human incidents in numerous forests all over Europe. Realizing the demand for an efficient and sustainable solution for the problem, the project was funded by the European Space Agency's Business Incubation Center (ESA BIC). With access to the grant, the project was laid out to develop a functional prototype on which a business model can be based. The outcome of this work were two prototype models, the Hexacopter v1.0 and the Coaxial Octacopter v2.0.

The experienced gained from working with these models was used for the development of the new prototype design. The Hexacopter v1.0 passed all flight and applications tests successfully, and thereby confirmed the feasibility of project. To improve injection performance, the Octacopter v2.0 was equipped with a mechanical arm. However, it turned out that a moving arm has negative effects on the stability of the drone during operation. Furthermore, it was previously confirmed that the amount and type of pesticides used for the injection is sufficient to interfere with the PPM's reproductive cycle, but does not harm other species, branches or tree leaves, and is thereby eliminating the threat while conserving the environment.

This study elaborates on two new prototype models, the *Octacopter v2.1* (Figure 2.2a) and the *Quadcopter v1.0* (Figure 2.2b). As the name suggests, the Octacopter v2.1 is an enhanced version of the Octacopter v2.0. The work on this drone was performed by the team members of the TSA Center, therefore it will only be mentioned briefly and no detail on its development process are given. The *Quadcopter v1.0* however is a pilot production for this type of application. The hypothesis was, that the smaller size and better mobility could allow for a more precise injection process. Additionally, it could be studied whether the flight duration, or number of nest injection per battery pack could be increased. The work on this drone was com-



pletely performed by the student. The individual parts of the frame, the batteries, the flight controller, transmitter and receiver were available before hand, but had to be assembled. The motors, propellers, ESCs and camera had to be selected according to their properties and ordered subsequently. This process will be described in the following sections.

## 2.2 Mission design

This section describes the mission procedure in more detail. Below a list of mission segments and in Figure 2.1 an illustration of the mission work flow can be found.

In the final stages of this project, the mission to be conducted by the drone fleet will be split into 8 different segments:

1. Set up a ground station and prepare the drones for operation.
2. Use a fixed-wing drone model to overfly the area and georeference trees with high accumulation of nests, also called 'hot-spots'.
3. Upload the reference map into the *cloud*.
4. Injection drone-fleet accesses the map via the *cloud*.
5. Drone-fleet takes off. It uses autonomous way-point navigation to choose its flight pattern.
6. Once a drone spots a nest, it initializes the approach and injects the pesticides. After successful completion it distances itself from nest again and prepares the same steps for the next nest.
7. The sprayed nests and covered area are tracked on the map.
8. The injection drone fleet returns to ground station to recharge batteries and refill the pesticide tanks.

Adding to that, a minimum of two individuals will be present at the ground station; one to supervise the operation, to ensure a safe and correct procedure, and another to exchange batteries and refill the pesticide tanks in Step 8. The autonomous way-point navigation is intended to ensure the shortest route between trees in order to save valuable flight time. Furthermore, only one drone shall be appointed per tree to avoid system interference and/or physical contact.

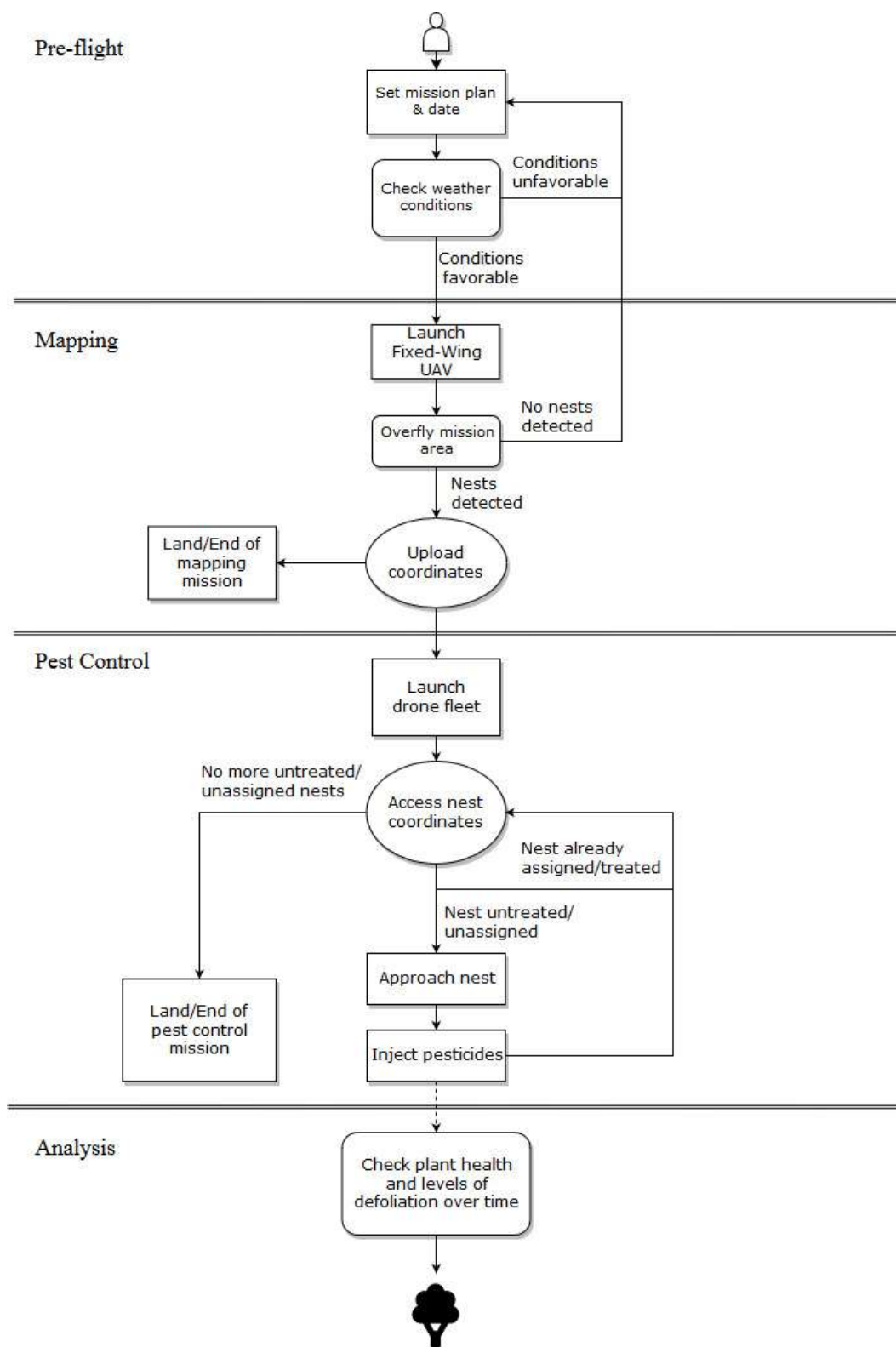


Figure 2.1: Mission work-flow diagram.

## 2.3 Hardware design

As mentioned previously, two drone models were developed simultaneously for the sake of this project. The models are:

1. Octacopter v2.1 (Figure 2.2a)
2. Quadcopter v1.0 (Figure 2.2b)



(a) Octacopter v2.1.



(b) Quadcopter v1.0.

Figure 2.2: Prototype models.

### 2.3.1 Prototype comparison

Table 2.1 presents the different segments and subsystem of the vehicle. The first column appoints a number to each part of the system. The indicator can be found in Figure 2.3 accordingly. This will be the last mentioning and detail on the *Octacopter v2.1*. The rest of the document will focus on the development of the *Quadcopter v1.0*.

Table 2.1: Prototype comparison

#	Segment	Octacopter v2.1	Quadcopter v1.0
1	Frame	RO QUAD V2 Foldable/X8	DJI F450
2	Material	Carbon fiber	Fibreglass composite
3	Motors	T-Motor MN4006 Antigravity Brushless Motors -380KV	TMotor F40 PRO II - 1600KV
4	Propellers	Carbon fiber 16x5.5 Multirotor	GEMFAN Flash 7042-2 Durable
5	Controller	Pixhawk PX4	Pixhawk PX4
6	Dimensions	56x56x40cm	37x37x8cm
7	OEW	3kg	822g



Figure 2.3: CAD model of FSD with part index according to Table 2.1.

### 2.3.2 Hardware component selection

Understanding the properties of the different system components was a crucial aspect when designing the drone. This section highlights the component's specifications and shows the reasoning behind their selection.

#### Frame

The *DJI Flamewheel F450*, seen in Figure 2.4, has an extremely smart design. The arms have a honeycomb like feature, which makes the structure sturdy and shock resistant while maintaining a low weight. It is fairly easy to assemble and the wide variety of spare parts make it the perfect choice for a prototype that is likely to experience several hard landings during its testing phase. The hole features in its arms also result in a better resistance against unfavorable weather conditions, such as strong winds or sudden gusts. It comes at an affordable price and its striking colors make it visible at long range.

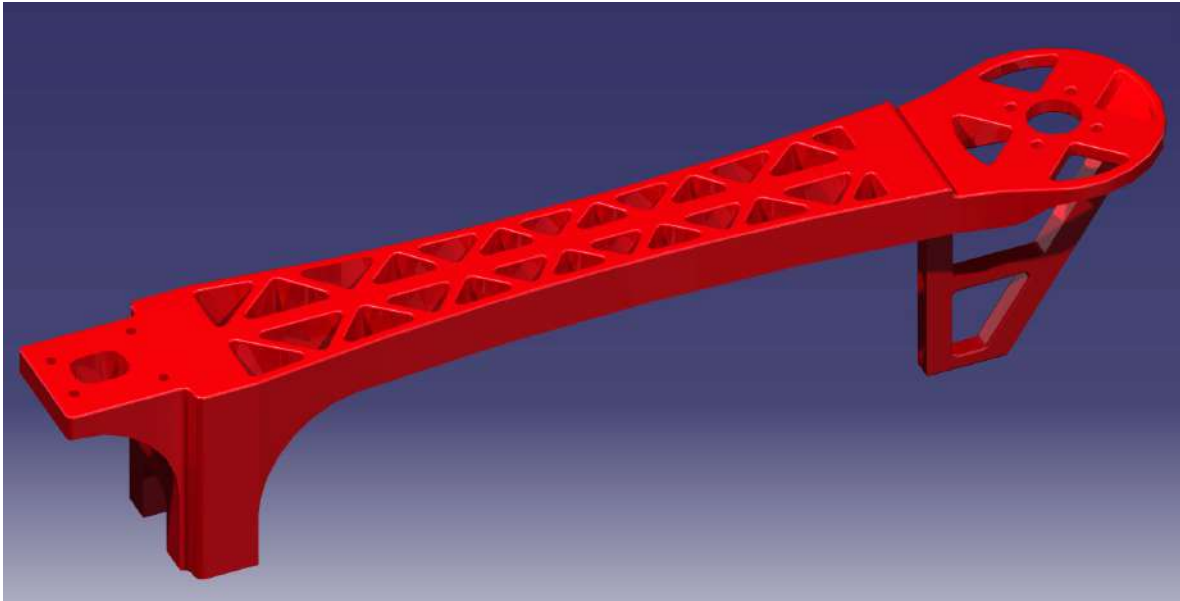


Figure 2.4: CAD model of DJI Flamewheel F450 frame part.

#### Propellers

Based on the size of the frame, the diameter of the propeller could be determined. The *DJI Flamewheel F450* is 450mm in diameter. The image in Figure 2.5 was used to set up Equation 2.1. Here  $X$  refers to the width and length of the frame, while  $D$  stands for the maximum propeller diameter.

$$\frac{D}{2} = \frac{X}{2} \quad (2.1)$$

So for a wheelbase of 450mm, measured diagonally from one motor position to another, using Pythagoras's theorem, width  $X$  is found to be 318.2 mm (or 12.53 inches). Including a safety margin of 50 mm (or 2 inches) to account for turbulence and to avoid propeller collision, a suitable propeller size was 10 inches. This also accords with the recommendations given by *DJI* [15].

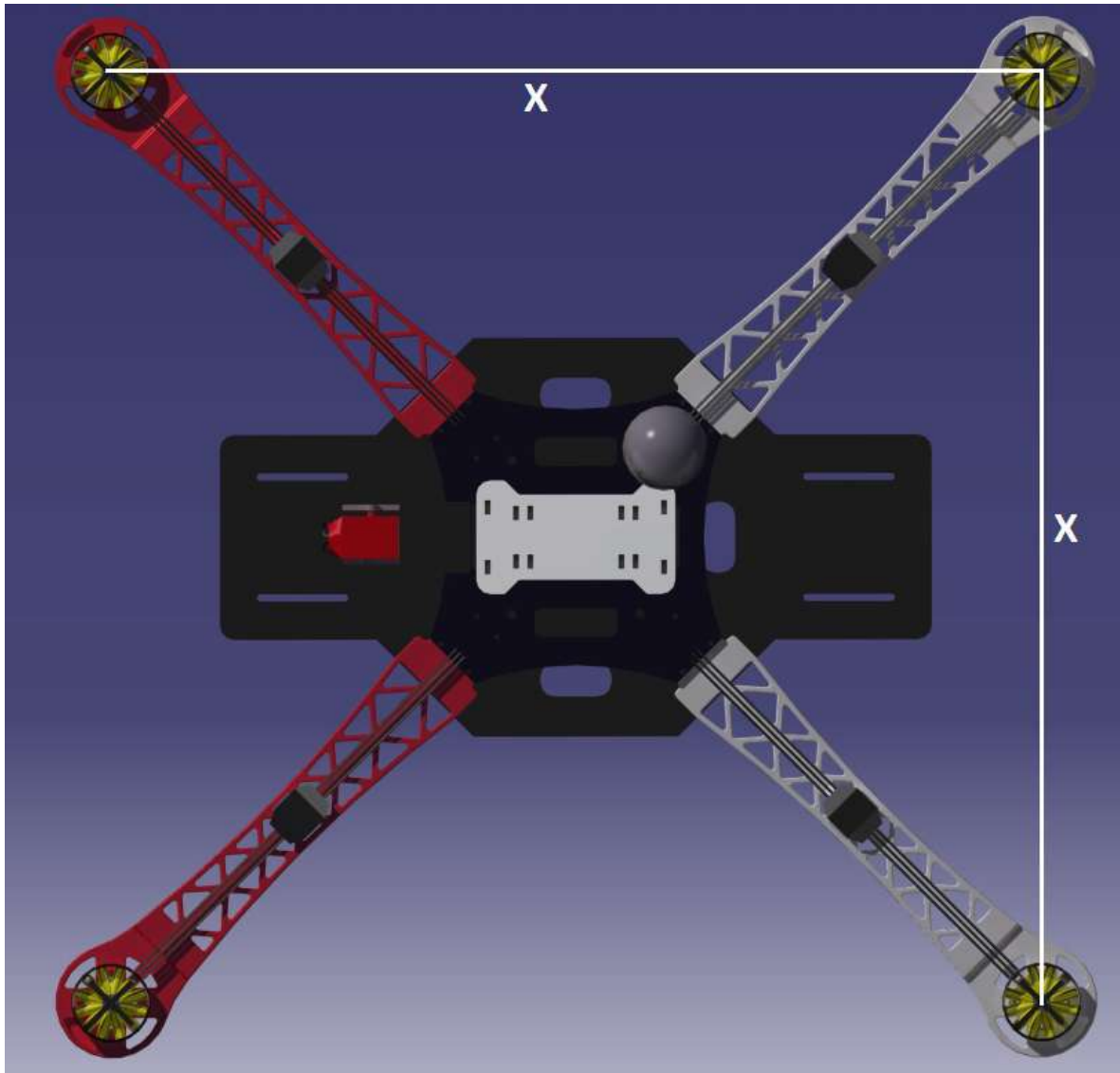


Figure 2.5: Top view of FSD.



## Motors

To make an appropriate motor selection, it was important to understand the nature of its application. In our case, the motors needed a high thrust-to-power(-consumption) ratio to achieve the longest possible flight time, resulting in more nests to be treated with one battery pack. Since the drone does not have to perform fast maneuvers, the power itself was less impacting in the selection process. However, a rule of thumb is to have at least a 4:1 power to weight ratio [16]. The ratio was calculated by taking the maximum thrust  $T_{tot}$  of the four *T-MOTOR F40 PRO II 1600kv* motors and divide it by the estimated total weight  $W$ .

$$P_r = \frac{T_{tot}}{W} = \frac{4 \cdot 1684g}{822g} = 8.19 \quad (2.2)$$

This exceeds the required thrust-to-power ratio leading to a fast control input response and an agile flying performance.

Item No.	Prop	Throttle	Thrust (g)	Voltage (V)	Current (A)	RPM	Power (W)	Efficiency (g/W)	Operating Temperature
F40proII KV1600	DAL6040 2-blade	55%	643	23.86	7.63	17169	182.14	3.53	99°C (Ambient Temperature : 15°C)
		65%	843	23.93	11.44	19430	273.74	3.08	
		75%	1059	23.89	16.12	21720	385.13	2.75	
		85%	1293	23.80	21.54	23643	512.64	2.52	
		90%	1386	23.76	24.26	24442	576.44	2.41	
		100%	1684	23.46	31.99	26745	750.49	2.24	

Figure 2.6: T-Motor F40 Pro-II thrust table [22].

## Controller

With an open source soft- and firmware, the *Pixhawk PX4* flight controller is probably one of the best flight controllers for autonomous flight on the current market [18]. It is equipped with a powerful 32-bit processor with additional backup controller and extensive memory [19]. It is simple to set up since most peripherals are automatically detected and configured, like an external GPS or other sensors.

## Batteries

As mentioned in Subsection 1.2.6, the flight duration is one of the limiting factors in any drone application. Therefore, special attention had to be paid when it came to selecting the right type of battery.

Lithium Polymer based (LiPo) batteries are being used for the majority of R/C aircraft and drones nowadays [20]. Their high energy density allows them to store more energy at a lower weight compared to other types of batteries. The two main possible alternatives to LiPo batteries are Nickel Metal Hydride (NiMH) and Nickel Cadmium (NiCd) batteries [21]. However, significant advantages that LiPo batteries offer are:

- LiPo batteries have higher power storage capacities.
- LiPo batteries have higher discharge rates allowing faster power transfer.
- LiPo batteries are lighter and come in different shapes and sizes.

The disadvantages of LiPo batteries are a shorter lifespan (300-400 cycles) compared to NiMH and NiCd batteries. Furthermore, there is a risk of fire and chemicals release if the cells get severely damaged. This requires some extra care when charging, discharging, or storing LiPo batteries.

After it was decided to use LiPos for the FSD, the next step was to select a battery pack with appropriate discharge rate and capacity. If the discharge rate (C-Rating) is too low, the battery does not transfer current fast enough to the motors, resulting in the drone not performing at its full potential [20]. However, increasing the C-Rating also increases the weight. Therefore, choosing a C-Rating that is too high can ultimately reduce the flight time due to the extra weight that needs to be carried.

$$DC[A] = C_{Rating}[C] \cdot Capacity[Ah] \quad (2.3)$$

Equation 2.3 is a simple equation to determine the optimum battery properties in terms of C-Rating and Capacity. As the motors are the subsystem with the highest power consumption, the discharge current DC is commonly found by looking at thrust tables that are often provided by the manufacturer of the motor. In case of the T-Motor F40 ProII 1600kV, the discharge current per motor at 75% throttle in a comparable propeller configuration is 16.12A, see Figure 2.6. Multiplying this by four to account for all motors resulted in a total DC of 64.48A. The value for current consumption was taken at 75% and not at 100% because the drone is only flying at full throttle at lift-off or to correct its position during strong wind gusts. In hovering and cruise flight the drone is more likely to fly at 40-60% throttle [23]. To account for



these sudden peaks in current consumption, some batteries, like the two types by Turnigy, have two C-Rating indications, see Table 2.2. The higher value can only be sustained by the battery for a couple of seconds and is often call burst rating [24].

For this project the following three LiPo batteries were chosen and tested for their performance:

1. DJI Phantom 2200mAh 3S 20C Battery
2. Turnigy MultiStar 1400mAh 3S 40-80C Multi-Rotor LiPo Pack
3. Turnigy A-Spec 1800mAh 3S 65C-130C LiPo Pack

The following table was completed using the information given above and Equation 2.3.

Decription	C-Rating [C]	Capacity [Ah]	$DC_{max}$ [A]
DJI Phantom	20	2.2	<b>48</b>
Turnigy MultiStar	40-80	1.4	<b>56-112</b>
Turnigy A-Spec	65-130	1.8	<b>117-234</b>

Table 2.2: Maximum discharge current of selected batteries.

Table 2.2 shows that the battery by DJI is not suitable to sustain the current consumption by the system. The Turnigy MultiStar produces just slightly less than the required current. However, the drone is unlikely to require this high consumption for long a long period of time. Only the Turnigy A-spec can support the system's power consumption over the full duration of the mission without taking damage.

## 2.4 Software design

This section of the report will focus its attention on the development of the detection algorithm. It presents the approach taken for this task, the methods used and the final product in form of a software code.

### 2.4.1 Methodology

The development of the system software was split into different stages. Starting with a rather simple approach, the algorithm became more sophisticated as the project evolves. The task descriptions for the different stages are:

1. Detect from image.
2. Detect from video recording.
3. Detect in real-time.
4. Detect, then act<sup>1</sup> based on mouse-click event (pilot input).
5. Detect, then act autonomously.

This approach was used to ensure a working algorithm that could continuously be tested during the different stages of its development. Detecting the nests was done by means of visual colour detection. Since the nests are white in a green surrounding, this method suggested a promising result.

### 2.4.2 Color based object detection

The open-source library *OpenCV* was used for the initial testing because of its easy and free access as well as the vast community of users and pre-build functions. This section explains the method behind color based object detection using *OpenCV* and *Python*. It presents the different image processing functions inside *OpenCV*, such as color space conversion, color based segmentation, masks, morphological transformation and contours.

#### 2.4.2.1 Color space conversion and value range.

Before the actual color based detection of the nest was achieved, a side application was created to determine the color value range of the desired object.

---

<sup>1</sup>Here, the 'act' is to approximate the nest and inject the pesticides.

## Mouse-click events

A mouse callback function is executed when left, right or double mouse click are given as input. Here the event takes **cv2.EVENT\_LBUTTONDOWNCLK** as input, which corresponds to a double left mouse click. It then uses the coordinates of the clicked pixel to perform its function. For this project it was used to output the HSV values of the clicked pixel (see Listing 1). By doing so, the program can be 'trained' to detect pixels in this specific value range, which then is used to execute the functions that are explained in the following paragraphs.

```
def print_hsv(event,x,y,flags,param):  
    if event == cv2.EVENT_LBUTTONDOWNCLK:  
        print hsv[x,y]
```

Listing 1: Creating Mouseclick-events.

## Color space

Overall there are more than 150 color-space conversion methods available in OpenCV [25]. The two most commonly used conversions are BGR-to-GRAY and BGR-to-HSV; here the latter is used.

For the colour conversion the pre-build function **cv2.cvtColor(input\_image, flag)** is used. The flag determines the type of conversion, in this case the conversion from BGR-to-HSV, which in *OpenCV* it is defined as **cv2.COLOR\_BGR2HSV**. It has to be noted that for HSV the Hue range is [0,179], Saturation range is [0,255] and Value range is [0,255]. The conversion to HSV is especially useful because the color is defined by one channel (H), which makes it a useful tool for color detection.

```
#Load RGB image  
img = cv2.imread('nest1.jpg',cv2.IMREAD_COLOR)  
  
#Resize image to fit screen  
img2 = cv2.resize(img,(640,480))  
  
#Convert BGR to HSV  
hsv = cv2.cvtColor(img2, cv2.COLOR_BGR2HSV)
```

Listing 2: Performing color-space conversion.

## Masks

Object detection involves the separation of certain regions inside an image by identifying differences in their properties. Such properties can be contours, intensity or color. Since the objective is color-based object detection, the code uses the pre-build **cv2.inRange(image\_name,lower,upper)** function. As input it takes the image, which has previously been converted to HSV, and the *lower* and *upper* HSV values to be displayed in the masked image when the application is started. The limits have to be given in form of an three cell array, giving an H-value between [0,179], S-value between [0,255] and V-value between [0,255]. The source code outputs a black and white image (see Figure 2.7 bottom right), which is a useful visualization that allows separate the desired object from the original image using the trackbar application described in the next paragraph.

```
#Define initial value range to be displayed
lower = np.array([0,0,0])
upper = np.array([179,255,255])

#Display masked image with desired color spectrum
mask = cv2.inRange(hsv,lower,upper)
```

Listing 3: Creating a masked image with desired color spectrum.

## Trackbars

At this stage the code was able to convert the original RGB image into an HSV image and display HSV values of any desired region by double clicking that area. The next step was to create a block of code that allows the user to fine tune the HSV value range that will later be used by the code to detect the nests. A direct approach was to create so called trackbars using the **cv2.createTrackbar** function, see Figure 2.7 top right. Sliding the trackbar will simultaneously change the corresponding colors in the masked image.

```
#Create trackbars
cv2.createTrackbar('H min','Mask', lower[0], 179, nothing)
cv2.createTrackbar('H max','Mask', upper[0], 179, nothing)
cv2.createTrackbar('S min','Mask', lower[1], 255, nothing)
cv2.createTrackbar('S max','Mask', upper[1], 255, nothing)
cv2.createTrackbar('V min','Mask', lower[2], 255, nothing)
cv2.createTrackbar('V max','Mask', upper[2], 255, nothing)

while(1):
    #get current trackbar position to change display values
    lower[0] = cv2.getTrackbarPos('H min','Mask')
    upper[0] = cv2.getTrackbarPos('H max','Mask')
    lower[1] = cv2.getTrackbarPos('S min','Mask')
    upper[1] = cv2.getTrackbarPos('S max','Mask')
    lower[2] = cv2.getTrackbarPos('V min','Mask')
    upper[2] = cv2.getTrackbarPos('V max','Mask')

    #create new masked image with new upper/lower range and display
    mask = cv2.inRange(hsv,lower,upper)
    cv2.imshow('Mask',mask)
```

Listing 4: Creating trackbar application.

In the second block of code, the current trackbar position overwrites the initial lower and upper values that are used to display the masked image, thereby changing the masked image in real-time. This makes it an easy application to use very adaptable to a wide range of colors.

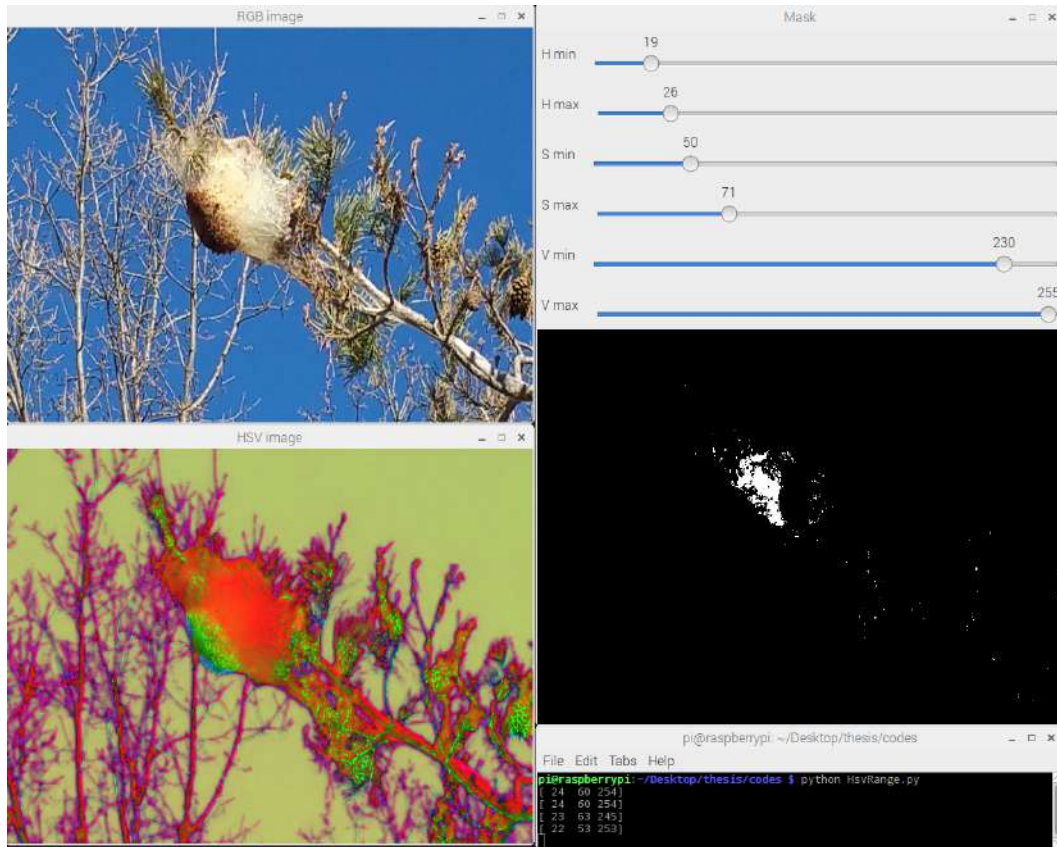


Figure 2.7: Application to determine HSV value range.

#### 2.4.2.2 Object detection from image.

We have previously learned about the application that helps the user to determine the HSV value range and the operations that are necessary for its functioning. This knowledge is now applied in an algorithm that is capable to detect and highlight the desired object. For this application, two additional operations have to be performed. At first, morphological transformations are used to remove noise, after which the contours of the object are detected and highlighted by drawing an enclosing circle.

#### Morphological transformations

Figure 2.7 shows some noise in the masked image that is not related to the nest but happens to have the same HSV values. So called Morphological transformations are used to remove such noise. It is an important procedure to ensure that no false-positive objects are displayed.

Morphological transformations are operations based on the object shape. It is usually performed on images in which the pixels have only two possible values (e.g 1 or 0), also called binary images. There are two basic ideas behind morphological operations, *Erosion* and *Dilation*.

During the operation of *Erosion*, the kernel, an area the size of a few pixels, moves along the masked image. Any pixel of the original image will only be considered as 1, if all the pixels under the kernel are 1 or else the pixel will be 'eroded' (made 0). This is very useful to remove white noise near the boundary of the object.

*Dilation* functions opposite to *Erosion*, here a pixel of the original image will be given the value 1, if only a single pixel from the masked image under the kernel has the value 1. Dilation usually follows Erosion, because during the process of Erosion some pixels inside the object might be removed. Performing Dilation brings these pixels back and thereby rejoins broken parts of the object.

The function used for this algorithm is **cv2.morphologyEx()**. This function initiates an operation called *Opening*, which is simply the process of Dilation followed by Erosion. As input the function takes the masked image and the kernel, which has to be defined by the user. It is followed by a *Closing* operation, which refers to Dilation followed by Erosion, used to close small holes in the image and to remove black pixels from the object.

```
#Perform morphological operation
kernel = np.ones((11,11),np.uint8)
morpho = cv2.morphologyEx(mask, cv2.MORPH_OPEN, kernel)
morpho = cv2.morphologyEx(mask, cv2.MORPH_CLOSE, kernel)
```

Listing 5: Applying morphological transformations.

## Contours

Drawing a contour is simply performed by joining all continuous points along a boundary of pixels with the same color and intensity. Therefore, binary images should be used also here for accurate results. The function that is being used to find the contours of the object is **cv2.findContours()**. As input it requires the image that has previously cleaned from all the noise, the contour retrieval mode and the contour approximation method. The contour retrieval mode retrieves all of the contours and reconstructs a full hierarchy of nested contours. The contour approximation method is used to store the location coordinates of the pixels that make up the contour of the desired object. The function outputs the contours of the object and the hierarchy, which specifies the relationship of how contours are connected.

Subsequently, the contour is drawn using **cv2.drawContours()**. As input it takes the source image, the list of contours passed by the previous function, and the index of contours (-1 is used to include all contours). The remaining arguments are used to define colour, thickness and style. The final block code that draws and outputs the



```
#Find contours  
im2, contours, hierarchy = cv2.findContours(morpho, cv2.RETR_TREE,  
↳ cv2.CHAIN_APPROX_SIMPLE)
```

Listing 6: Finding contours within the image.

image with detected object is:

```
#Draw contours  
cv2.drawContours(draw, contours, -1 (0, 255, 0), 2)  
cv2.imshow('Contours image', draw)
```

Listing 7: Drawing contours around detected object.



Figure 2.8: Image with colour based object detection.



### 2.4.2.3 Object detection from video recording

The next step in the software design was to adjust the previous code to be able to detect the nests from a video recording. Since the previous code successfully detected the object from a single image, the approach for detecting objects from a video was the following: Each image frame of the video recording was selected individually and the same operations as described previously were performed. In order to read a video file into the code, the function **cv2.VideoCapture()** was used. It takes the name of the video file as its only input. Note that the file has to be in the same directory as the Python file itself. Once the video file is opened, the algorithm enters a loop in which it individually takes every single frame to perform the above mentioned operations.

```
#Input video file
capture = cv2.VideoCapture('video.mov')

#select individual frames
while(capture.isOpened()):
    ret,image = cap.read()

    #at this stage perform color conversion, morphological
    → transformation etc. on single frame

    #display the frame
    cv2.imshow('Frame', image)
```

Listing 8: Read video recording into detection algorithm.

Running the code resulted into the detection of many false-positive objects, see Figure 2.9a. These objects are almost exclusively very large, like the trunk of the tree, or very small, such as a spot of snow on a branch. In order to compensate for this, a size filter was applied to the existing bit of code that draws the contour around the object. The improvements this filter resulted in can be seen in Figure 2.9b. The size range was determined by trial and error.

```
#only display contours of certain size
if radius > 35 and radius < 65:
    cv2.circle(image, center, radius, (0, 0, 255),2)
```

Listing 9: Apply filter to remove false-positive objects.



(a) Nest detection before improvement.



(b) Nest detection after improvement.

Figure 2.9: Nest detection from video recording.

#### 2.4.2.4 Object detection in real-time

Two different approaches were taken for the task of detecting the nests in real-time. At first, the platform RaspberryPi (RPI) with build-in camera module was used. In a later stage, a combination of on-board camera and video transmitter, ground video receiver and analog-to-digital converter was used to transfer the video feed from the FSD to the ground station, where all further operations were performed.

##### On-board

At first, the camera inside the RPI is accessed using the pre-build **PiCamera()** function. Then its properties were defined. The software uses the raw footage at an resolution of 640x480 pixels and a frame rate of 32 fps. After the camera is given some time to warm up (**time.sleep(0.1)**), the frames that are captured by the PiCam are accessed individually using **camera.capture\_continuous(rawCapture, format='bgr', use\_video\_port=True)**. It outputs a numpy array that represents the image. This image is taken to perform all perviously explained operations to detect and highlight the nests.

```
#start up camera and grab raw camera capture
camera = PiCamera()
camera.resolution = (640, 480)
camera.framerate = 32
rawCapture = PiRGBArray(camera, size=(640, 480))

#allow camera to warm up
time.sleep(0.1)

#capture camera frames
for frame in camera.capture_continuous(rawCapture, format='bgr',
    ↪ use_video_port=True):

#grab numpy array representation of image
image = frame.array

#now perform all previous operations on the image
```

## Ground-station

Configuring and installing a functioning ground-station was a difficult process that often lead into a dead end. However, attaching the RPI on the drone was not a suitable solution due to its size, power consumption and bad image resolution. To overcome this hurdle, a small FPV camera and video transmitter were installed on the drone platform. The signal was received on the ground-station by using a *Duo5800 – 5.8GHz A/V Diversity Receiver*. To test this set-up, the receiver was connected to a basic portable LCD display. The set-up was verified as the screen displayed a real-time colored image of the FPV camera. The next step was to read the video into the previously described algorithm for object detection. A similar method as previously used for the detection from a video recording was considered. The pre-build `cv2.VideoCapture()` function has the ability to read frames from an external web-cam if given the right index (in this case the index was 0). However, a crucial aspect that was not considered is that the video receiver outputs an analog video signal. Unfortunately, the above mentioned function is not capable to understand analog video signals. After some research, it was found out that the transformation of the video into digital form could be done via a specific adapter, called *EasyCap*. After some initial difficulties related to the driver installment, the video transmitted by the subsystems on the drone was successfully displayed on the ground station in real-time, see Figure 2.10.

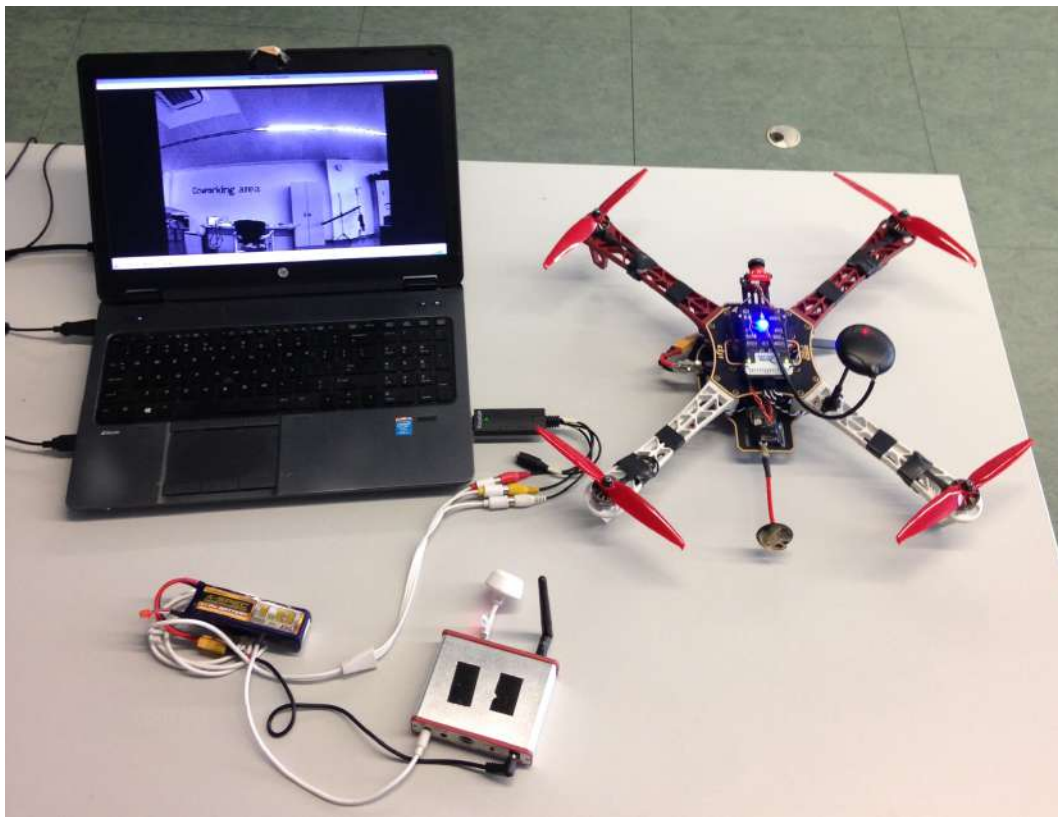


Figure 2.10: Test set-up for video transmission to ground station.



## 2.5 Testing the system

With all parts assembled and working individually, it was time to put the system to a test. A proven testing ground for the FSD over the past two years has been the area of Serra de Montgrony, Spain. This mountain range in the north of Catalunya is home to big deposit of pine trees and is located an acceptable 120km away from the companies head quarter in Cerdanyola de Valles. It is close enough to preform recurring tests and follow-up checks on the PPM population, while being out of the distance of populated areas resulting into the flight tests following FAA regulations.



Figure 2.11: Pine tree in Serra de Montgrony with 15+ nests.

## 2.5.1 Testing the hardware

The system was tested individually for its flying performance and for its detection algorithm. As previously mentioned in Section 2.3, there are actually two prototypes of the FSD being developed simultaneously. Due to its proven design, the *Octacopter v2.1* was used to test the hardware on its functionality during the injection of pesticides directly into the nests.

### Flight duration per battery pack

To assess whether the system will be capable of performing a large scale operation, as in treating a large area of pine trees, the system was tested for its flight duration when performing the injection process. The total flight duration per battery pack was measured as well as the time it takes to perform the pesticide injection into a single nest. The tests were performed as follows:

1. A fresh battery pack was attached to the *Octacopter v2.1*.
2. At lift-off, the first timer was started.
3. When the first injection attempt was initiated, the second timer was started. (It sometimes took more than one approach to inject the needle to the nest)
4. After successful injection, the second timer was stopped.
5. The drone was guided to the closest untreated nest to repeat the last two steps.
6. When the power stored inside the battery was used up, the drone was landed, and the first timer was stopped.

The tests resulted into the following data, see Table 2.3.

Table 2.3: Battery-life test data for *Octacopter v2.1*

# Battery	Total flight duration [s]	# Injections	Avg. time per injection [s]
1	11:45	2	01:36
2	07:13	3	00:56
3	09:37	1	01:10
4	10:53	1	01:17
5	11:21	3	01:48

## Injection process

At this stage of the project the drone was still being operated by a pilot. The testing of the injection process was performed to see if the weight components of the system were balanced properly, allowing for good stability and control. Furthermore, the flight tests were able to show whether the structure can withstand the loads that are present during contact between injection arm and nest and during possible collision with surrounding branches.

The following observations were made:

1. The drone was easy to control by the pilot. The reactions to control inputs (thrust, roll, pitch, yaw) were direct and smooth.
2. The drone was able to maintain constant height when set to altitude-hold mode.
3. Despite in-situ compass calibration, the drone displayed a constant drifting behaviour when set to position-hold mode.
4. The needle attached to the tip of the injection arm was broken loose several times after contact with the nest.
5. The drone structure and propellers were able to withstand repeated collision with surrounding branches without loss of performance.

There was an additional, very crucial observation. Positioning the drone in very close vicinity to the nest did not cause any difficulties. However, for the drone to move forward, it had to slightly tilt around its pitch-axis. Even if the pitch angle was small, it created a great enough displacement over the length of the injection arm for the needle to regularly miss its target. This is an important observation and has to be taken into account when developing the autopilot in later stages of the project.

## 2.5.2 Testing the software

The software algorithm was tested in two ways. At first, the algorithm used to 'detect from video recording' was tested by recording the nests using the high resolution camera on board of the *DJI Inspire 2*. Afterwards, the algorithm to 'detect in real-time' was tested on the nests located on the bottom branches using the RPI. At the time these tests were performed, the system configuration using the on-board camera, ground video receiver and EasyCap converter were not ready for testing. This set-up was tested at a later stage in a laboratory environment.

### Detect from video recording

This was the first time the software algorithm was tested on a video recording of nests located in the actual area of operation. Up to this point, the software has only been trained using the trackbar application described in Subsection 2.4.2. The

outcome of this test resulted into a smooth video that displayed contours around the nests. A screenshot of this video can be found in Figure 2.9b. The following observations were made from this test:

1. The video displayed several false-positive objects.
2. The video was returned at a very low frame rate (fps).

Based on these observations, the adjustments to the code as found in Listing 9 were made, resulting into a video with almost no false-positive object. The frame rate of the video was very dependent on the platform the algorithm was ran on. On the RPI, the frames rate was considerably lower than on a laptop. Also, running the algorithm several times on the same platform still resulted in differences in frame rate.

### **Detect in real-time using RPI**

Since attaching the RPI to the drone platform was not a suitable solution at the time, this test was performed manually via hand-holding the device. The RPI was connected to the ground-station via WiFi-hotspot connection. The camera was then pointed at nests that hang on the lower branches of the pine tree. Care was taken to mask the motion of a hovering drone during this test. The following observations were made:

1. The algorithm was able to detect the nests, despite the reduced image resolution.
2. The frame rate of the video output was identical to the frame rate when accessing the PiCamera without all the extra frame operations.

The video was expected to be saved using the `cv2.imwrite()` function. Unfortunately, the algorithm did not output a saved video.

### **Detect in real-time with ground-station**

As mentioned previously, at the time of the field test, the ground-station set-up using the on-board camera, video receiver and EasyCap device were not ready for testing. This configuration was tested at a later stage in the drone laboratory of the Universitat Politècnica de Catalunya, Spain. To test this set-up, the drone was armed and located in the center of the flying area. The ground-station was equipped with the EasyCap device and connected to the video receiver, see Figure 2.10. After the video was successfully displayed on the ground-station, the drone was armed and taken to the air. The following observations were made:

1. The algorithm and its build-in function were not able to detect the EasyCap device as external webcam.
2. The video stream often had disturbances when Tx antenna was pointed in opposite direction of Rx.



## Chapter 3. Evaluation

This chapter contains an evaluation of the field test results described in Section 2.5. Achievements and importance of this work are highlighted here, points of failure are analyzed and possible methods for improvement are suggested.

### 3.1 Hardware

This section deals with the evaluation of the hardware tests performed with the *Octacopter v2.1*.

#### Flight performance

The tests performed to evaluate the flight performance of the *Octacopter v2.1* signaled the following *strengths*:

1. The drone was easy to control by the pilot. The reactions to control inputs (thrust, roll, pitch, yaw) were direct and smooth.
2. The drone was able to maintain constant height when set to altitude-hold mode.
3. The drone structure and propellers were able to withstand repeated collision with surrounding branches without loss of performance
4. The drone was able to lift the battery pack and injection payload and still performed well in terms of mobility, stability and flight duration.
5. The flight duration was long enough to perform the injection to at least one nest.

The tests have also highlighted some of the system's *weaknesses*, such as:

1. Despite in-situ compass calibration, the drone displayed a constant drifting behaviour when set to position-hold mode.
2. The needle attached to the tip of the injection arm was broken loose several times after contact with the nest.
3. Introducing a pitch angle to approach the nest resulted into a needle displacement great enough to repeatedly miss the target.

## Strengths

The strengths the *Octacopter v2.1* displayed during the tests will be evaluated in one or two statements, while the weaknesses will be analyzed in more detail in the following paragraphs.

Overall the drone did very well in terms of flight performance. This result was to be expected, since the different subsystems have been chosen carefully for this purpose and the individual parts have been tested previously by manufacturers and a wide range of consumers. It was proven during the tests, that the hardware is capable of performing the required operations to a satisfactory extend. The lifting, control and battery-life properties of the *Octacopter v2.1* make this prototype a suitable selection for further development on this project.

## Weaknesses

### 1. GPS inaccuracy

This observation was traced back to common problems in GPS location. To achieve an accurate 3D fix, the receiver that is mounted to the drone needs to be in direct line of sight with ideally six or more satellites. The test environment in Serra de Montgrony is full of high mountain ranges and tall pine trees. Therefore, the issues that were experienced when setting the drone to position-hold mode have their origin either in loss of direct line of sight or in so called Multi-path signals, which are signals that are reflected by the surrounding landscape that also cause inaccuracy in GPS data.

A possible solution to this problem can be the use of so called GNSS receivers. These receivers access position data not only from GPS satellites, but also from other global navigation satellite systems such as GLONASS, Galileo, BeiDou and more, thereby increasing the number of visible satellites in the sky.

### 2. Structural flaws

This problem can easily be solved by designing a more elaborate mounting system for the needle.

### 3. Tip displacement

Special attention has to be paid to this observation when designing the autonomous injection process. In order to compensate for the tip displacement of the injection arm, the following mathematical model can be used.

When looking at Figure 3.1, a correlation between pitch-angle and tip displacement can be derived. To do so, one has to realize that if the drone pitches at an

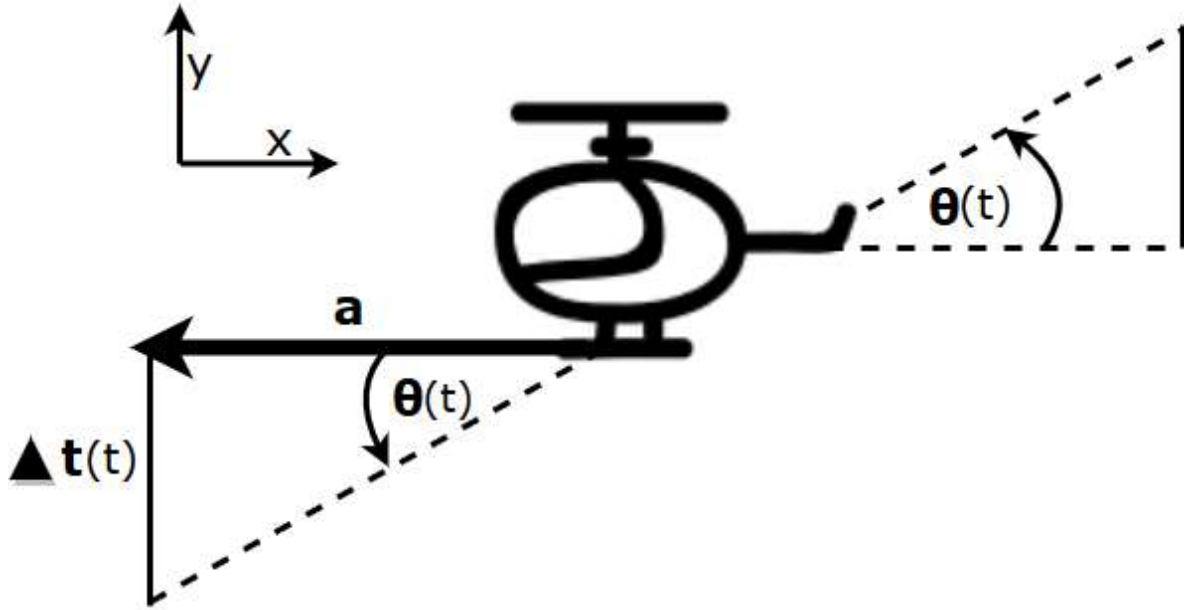


Figure 3.1: Correlation of tip displacement and pitch angle.

angle  $\theta$ , the injection arm is being displaced by the same angle. When the length from the center of the drone to the tip of the injection arm  $a$  is known, the tip displacement  $\Delta t$  can be calculated using Pythagoras's theorem, see Equation 3.1.

$$\Delta t = \frac{a}{\tan(\theta)} \quad (3.1)$$

To ensure that the tip of the injection arm hits the target at any given time of the injection process, the drone has to introduce a vertical displacement  $\Delta y$  that is equal to the displacement  $\Delta t$  at any given time  $t$ .

## 3.2 Software

This section deals with the evaluation of the software tests performed with the *Quadcopter v1*.

### Detection algorithm

The tests performed to evaluate the detection algorithm were performed in three different settings, as described previously in Section 2.5. The tests signaled the following *strengths*.

1. The algorithm was able to detect the nests, despite reduced image resolution

2. The algorithm was able to detect the nests, despite the different lighting condition

The tests have also highlighted some of the system's *weaknesses*, such as:

1. The video displayed several false-positive objects.
2. The video was returned at a very low frame rate (fps).
3. The algorithm and its build-in function were not able to detect the EasyCap device as external webcam.
4. The video stream often had disturbances when Tx antenna was pointed in opposite direction of Rx.

### Strengths

The strengths the detection algorithm displayed during the tests will be evaluated in one or two statements, while the weaknesses will be analyzed in more detail in the following paragraphs.

Considering the fact that the tests described in the previous section were the first displayed an overall good performance. Since previously the software has only been trained using the image operations described in Section 2.4, the impact that a change in lighting conditions and differences in image resolution has on the algorithm were unknown prior to the test. It turned out that these factors were less impacting than expected; The nests were successfully detected by the algorithm in each of the test settings. This implies that color based object detection is a suitable choice for the task laid out for this project.

### Weaknesses

Being the first software prototype, the algorithm displayed a handful of weaknesses during the testing.

#### 1. False-positives

It is a big challenge to design an object detection application capable of competing with modern face and object detectors used by successful companies like Apple, Snapchat and Facebook [26]. To diminish the detection of false-positive objects, further object classifiers would have to be added to the code. Possible classifiers for this application could be shape-based, texture-based or gradient-based. However, such operations require high computational costs, a resource that was not available for the development of this prototype. Therefore, it can be concluded, that although color alone is not a flawless measure for the detection and tracking of objects, the low computational cost of the color based algorithms make it a useful feature to be

exploited.

## 2. Low frame rate (fps)

A low frame rate is more likely to be classified as general observation, rather than a weakness. Since the objects to be detected by the algorithm are not in motion, a low frame rate has no direct impact on its performance. Since the drone is moving at a relatively low speed, the object will still be visible in several frames, long enough for the pilot to spot the nest, and at later stages to geo-locate the nest for the autonomous injection process.

## 3. Component incompatibility

This problem that the EasyCap device was not detected by the algorithm can be traced back to the Operating System (OS) used for the development of this application. The algorithm runs inside the Linux OS, however, the computer is based on Windows. To compensate for this, an emulation of a computer system was created using a virtual machine called VirtualBox by Oracle. Unfortunately, the EasyCap device is not compatible with virtual machines and emulators according to its user manual [27]. A computer platform running on a Linux OS would be a direct solution for this problem.

## 4. Tx/Rx disturbances

This problem can be traced back to the hardware components used for the application. The camera and video transmitter are designed for First-Person-View [FPV] flights, mostly used by hobby drone pilots for racing applications. The same day this system was tested, the *DJI inspire* produced beautiful video imagery, indicating that the disturbances are purely component and not environmental or software related. Hence, this issue can easily be removed by adding a more powerful TX/RX system that is suitable for this application. .

## Chapter 4. Discussion

This discussion will highlight the impact this research has on sustainability and the ethical question that arose over the course of this project. The chapter ends with recommendations for future research.

### 4.1 Sustainability

The impact this research has on sustainability will be analyzed with respect to ecological and economical aspects.

#### Ecological impact

Drones have long been considered as a leap in the direction of sustainability. Common sense suggests that using a vehicle powered by small electrical engines has less to no impact on its surrounding environment when compared to a Van or an Airplane. But does a drone beat a fossil-fuel powered vehicle when it comes to a global scale? A study by *Forbes* has produced a possible answer. The study revealed that drones use up twice as much energy then a standard Ford Transit when used at *Amazon* for package delivery [28]. However, drones have an unlucky draw in this study, since the applications involves carrying heavy payloads for a long period of time, both of which are main limitations in modern drone technology. Therefore, a second study has been consulted. This time the result seemed more accurate to this specific application. The study compared the overall emission of a vehicle powered by an electrical engine with a vehicle powered by combustion engine. The result was, that when taking into account the emission during manufacturing and the use of the electrical grid, the ecological footprint of an electrical vehicle is still better than the use of a combustion engine. Meaning that the use drones for the control of PPM outbreaks, rather than the traditional aerial spraying methods that make use of airplanes, has a *positive impact* to the environment in terms of emission. Furthermore, as mentioned in Subsection 1.2.3, the use of drones also eliminates the collateral damage caused to trees and wildlife during traditional aerial spraying methods and heavily reduces the risk for human operators.

#### Economical impact

From a ecological perspective, UAVs are considerably cheaper to purchase and operate than conventional aircraft used in aerial pest control operations [8]. However, the installation, configuration and maintenance of the system requires the advanced knowledge of technicians or engineers, which on the long term increases the total cost [11]. Considering the original applications of drones, such as surveying, mapping and inspection, the structure of commercial drones are not designed to carry heavy payloads such as pesticide tanks. Therefore, it is likely that modifications to the structure are necessary to accommodate the new payloads. Furthermore, being

subject to FAA regulations, agricultural drones need to be operated by licensed pilots, indicating another field to allocate resources to. Nonetheless, considering the € 1.0 to € 1.5 million that are spent annually on aerial spraying to control PPM outbreaks in Andalusia, the above mentioned costs are a small price to pay considering the immense reduction of operational costs. And either way, considering the failure of current pest control methods as mentioned in Subsection 1.2.3, and the success of the FSD concept, it is an alternative worth its investment.

## 4.2 Ethics

Following the principles for an ethical wildlife control is of special importance for this project, since it involves lethal controlling of the animal for the preservation of public healthy and safety. A study on behalf of the *Society for Conservation Biology* states that the following seven questions have to be answered to ensure ethical wildlife control [29]:

### 1. Can the problem be mitigated by changing human behavior?

No, the problem is a direct result of a change in climate conditions as mentioned in Section 1.2. If human behaviour causes the change in climate conditions is for the reader to decide.

### 2. Are the harms serious enough to warrant wildlife control?

Yes, the overpopulation of PPMs has a serious effect on wildlife and forest vegetation. The constant defoliation of pine trees strongly reduces pine growth as well as all the reproductive parameters such as the number, size and weight of the seeds and cones [30], simultaneously also reducing food for other forest inhabitants such as squirrels and birds. Furthermore, there is a direct threat to humans and dogs that visit the forest. Contact with the PPM causes skin rashes on humans and in the worst case leads to the death of their pet, if the swelling blocks the breathing canal [31].

### 3. Is the desired outcome clear and achievable, and will it be monitored?

Yes, the outcome is to control PPM population numbers to a point that the ecosystem can handle the threat by itself. The frequent monitoring of past and present distribution of PPMs in Europe, Asia and North Africa goes back to the 1970s and will continue in the future [2].

### 4. Does the proposed method carry the least animal welfare cost?

Yes, as mentioned in Subsection 1.2.5, the FSD concept has strong beneficial effects in terms of collateral damage when compared to traditional methods. Directly inserting the pesticides into the nests is a surgical process that eliminates negative

impact on other wildlife and vegetation.

*5. Have community values been considered alongside practical information?*

Yes, as mentioned in Question 2., the purpose of the project is partly to protect wildlife and vegetation, and partly to ensure the health and well being of the community living or visiting the area. The control technique is designed to operate in a predefined area roaster to ensure an efficient treatment of the problem.

*6. Are the decisions based on the situation rather than negative labels of the animals?*

Yes, the research and actions are purely performed because of the negative effects of the insect population on wildlife and vegetation.

### **4.3 Future Research**

Several option to further develop the FSD concept are suggested in the body of this section. These option include the direct improvement of the current system as well as recommendation to steer this project into new directions.

Developing a working autopilot is an important next step to make the system more reliable and automated, resulting into an overall more efficient operation. Dividing the area into smaller sectors, including only a couple of infected trees, could be a good starting point for the development, allowing for a learning process through trial and error. Alternatively, hiring and training pilots allows a faster injection and good area coverage, ensuring an actual impact to the problem.

Furthermore, it should be investigated whether the use of additional drones has a beneficial effect. Based on the recorded flight time and time it takes to inject pesticides into a single nest, one drone alone is not capable to treat all of the infected pine stands. A drone fleet could cover a greater area, resulting into a faster control of the outbreak. On the long run, this is likely to save time and hence money.

Last but not least, the optimization of currently used system components should be considered. As the components used for the prototype drones are mostly “good enough” to test the project for its validity, using optimal components will significantly improve the performance of the system. This is a field that consumes most of the financial resources, but using state-of-the-art technology will allow this project to make a giant leap forward, making it more interesting and eligible for investment and funding.



## Chapter 5. Conclusion

The purpose of this thesis was to research if a drone based system is a feasible concept for the control of Pine Processionary Moth outbreaks in the Mediterranean woodlands. Initiating the thesis with a broad research on current pest control methods and state of the art technology, the concept's feasibility was tested by implementing the system as a prototype project. The prototype was design based on requirements derived from drone related literature as well as from the practical needs of the company and investors. The project included the complete design of the system structure and mission work flow, acquiring and installing the required subsystems, testing the system in practice and evaluating the results. The evaluation was done by comparing the system's performance against traditional control methods and by analyzing how well the system performed in the tasks it was designed for.

The results have validated that the drone based control method is both a feasible and working concept. Using this new system allows to reduce the mission costs and manual labor, while offering a sustainable approach with respect to the environment. Continuing to work on this project with the constantly evolving drone technology will show that the system can be used for a multitude of different applications and therefore its potential should be studied further in the future.

## Bibliography

- [1] Hódar, J.A. and Zamora and R. and Castro, J. and Baraza, E. ; *Feast and famine: Previous defoliation limiting survival of pine processionary caterpillar Thaumetopoea pityocampa in Scots pine Pinus sylvestris*, Acta Oecol , 2004, 26, 203–210.
- [2] Alain Roques et al, *Climate Warming and Past and Present Distribution of the Processionary Moths (Thaumetopoea spp.) in Europe, Asia Minor and North Africa*, Springer, Chapter 3, October 2014.
- [3] Adrián Cardil and Udayalakshmi Vepakomma and Lluís Brotons, *Article Assessing Pine Processionary Moth Defoliation Using Unmanned Aerial Systems*, Forests, 23 October 2017.
- [4] Luis Cayuela and José Antonio Hódar and Regino Zamora, *Is insecticide spraying a viable and cost-efficient management practice to control pine processionary moth in Mediterranean woodlands*, Elsevier, 26 February 2011.
- [5] Andrew Nixon, *Best Drones for Agriculture 2018: The Ultimate Buyer's Guide*, Bestdroneforthejob, 11 March 2018.
- [6] Michał Mazur et al. *Clarity from above: PwC global report on the commercial applications of drone technology*, PWC, May 2016.
- [7] DJI, <https://www.dji.com/es/mg-1>, 2018.
- [8] Craig G. Morley and James Broadley and Robin Hartley and David Herries and Duncan McMorran and Ian G. McLean *The potential of using Unmanned Aerial Vehicles (UAVs) for precision pest control of possums (Trichosurus vulpecula)*, RethinkingEcology, 2017.
- [9] Federal Aviation Administration, *Request a Part 107 Waiver or Operation in Controlled Airspace*, FAA, 31 May 2018.
- [10] SprayingDrone, *A Cost Effective Solution*, <http://sprayingdrone.com/>, 2017.
- [11] Yanbo Huang and Steven J. Thomson and W. Clint Hoffmann and Yubin Lan and Bradley K. Fritz, *Development and prospect of unmanned aerial vehicle technologies for agricultural production management*, Int J Agric & Biol Eng, September, 2013.
- [12] F. Zimmermann and C. Eling and L. Klingbeil and H. Kuhlmann, *Precise Positioning OF UAVS - Dealing With Challenging RTK-GPS Measurement Conditions During Automated UAV Flights*, Institute of Geodesy and Geoinformation, University of Bonn.
- [13] Federal Aviation Administration, *Drone Certification: A Step-by-Step Guide to FAA Part 107 for U.S. Commercial Drone Pilots*, FAA, 21 June, 2016.

- [14] TSA Center/FitoStinger, *<http://www.fitostinger.com/>*, 2017.
- [15] DJI, *<https://www.dji.com/es/flame-wheel-arf/spec>*, 2018.
- [16] Anthony Jacobs, *How to choose the right size motors & ESCs for your Drone, quadcopter, or Multirotor build.*, QuadQuestions, February 22, 2017.
- [17] Hobbyrc, *<https://www.hobbyrc.co.uk/t-motor-f40-pro-ii-1600kv-motor-grey>*, Hobbyrc, 2018.
- [18] V. Kadamatt, *Best flight controller for quadcopter [2018] : How do I choose?*, DroneYbee, February 15th, 2018
- [19] ArduPilot Dev Teamt, *Pixhawk Overview*, jDrones, 2016
- [20] Alex Elliott, *LiPo Batteries - How to choose the best battery for your drone*, DroneTrest, September 2015
- [21] DroneOmega, *The Advanced Quadcopter Battery Guide*, DroneOmega, January 2018
- [22] We are FPV, *Moteurs T-Motor F40 Pro II 2306 1600KV — Overview*, Youtube, March 2018
- [23] ArduPilot Dev Team, *Setting Throttle Mid (aka Hover Throttle)*, jDrones, 2016
- [24] Anthony Jacobs, *Battery C rating explained*, QuadQuestions, December 27, 2014
- [25] Esther Salami, *MED Payload Lecture Slides*, UPC, 2017
- [26] Bhumika Gupta et al., *Study on Object Detection using Open CV - Python*, International Journal of Computer Applications, March 2017.
- [27] *EasyCap User Manual*, EasyCap Products, Version 1.5.
- [28] Bhumika Gupta et al., *What Would The Environmental Impact Be If Amazon Delivered Everything Via Drone?*, Forbes, May 17, 2016.
- [29] Sara Dubois et al., *International consensus principles for ethical wildlife control*, Wiley Periodicals, November 14, 2016.
- [30] José A. Hódar and Jorge Castro and Regino Zamora, *Pine processionary caterpillar *Thaumetopoea pityocampa* as a new threat for relict Mediterranean Scots pine forests under climatic warming*, Elsevier, March 2003.
- [31] Niza ME and Ferreira RL et al., *Effects of pine processionary caterpillar *Thaumetopoea pityocampa* contact in dogs: 41 cases (2002-2006).*, Zoonoses Public Health., February 2012.

# **Appendices**



## Appendix A.

---

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

# Mouse-click event function
# print [H,S,V] value of the pixel you clicked
def print_hsv(event,x,y,flags,param):
    if event == cv2.EVENT_LBUTTONDBLCLK:
        print hsv[y,x]

# Define foo function
def nothing(x):
    pass

# Upload RGB image
img = cv2.imread('nest1.jpg',cv2.IMREAD_COLOR)
img2 = cv2.resize(img,(640, 480))
cv2.imshow('RGB image',img2)

# Convert BGR image to HSV image
hsv = cv2.cvtColor(img2, cv2.COLOR_BGR2HSV)
cv2.imshow('HSV image',hsv)

# Call print_hsv function when double mouse-click
cv2.setMouseCallback('HSV image', print_hsv)

# define initial range of white color in HSV
lower = np.array([0,0,100])
upper = np.array([35,45,255])

# Create trackbars for threshold range
# H range is [0,179], S range is [0,255] and V range is [0,255]
cv2.namedWindow('Mask')
cv2.createTrackbar('H min','Mask',lower[0],179,nothing)
cv2.createTrackbar('H max','Mask',upper[0],179,nothing)
cv2.createTrackbar('S min','Mask',lower[1],255,nothing)
cv2.createTrackbar('S max','Mask',upper[1],255,nothing)
cv2.createTrackbar('V min','Mask',lower[2],255,nothing)
cv2.createTrackbar('V max','Mask',upper[2],255,nothing)

while(1):
    # get current positions of trackbars
    lower[0] = cv2.getTrackbarPos('H min','Mask')
    upper[0] = cv2.getTrackbarPos('H max','Mask')
    lower[1] = cv2.getTrackbarPos('S min','Mask')
    upper[1] = cv2.getTrackbarPos('S max','Mask')
    lower[2] = cv2.getTrackbarPos('V min','Mask')
    upper[2] = cv2.getTrackbarPos('V max','Mask')

    # Threshold the HSV image to get only the desired color
    mask = cv2.inRange(hsv, lower, upper)
    cv2.imshow('Mask', mask)

    # Press ESC to exit
    if cv2.waitKey(1) & 0xFF == 27:
        break

cv2.waitKey()
cv2.destroyAllWindows()
```

Figure A.1: Code to determine HSV range

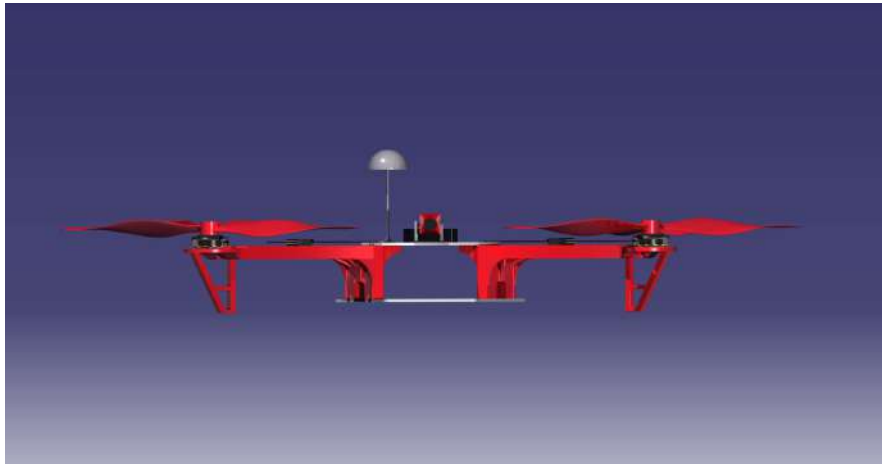
```

2 from picamera import PiCamera
3 import time
4 import cv2
5 import numpy as np
6 # initialize the camera and grab a reference to the raw camera capture
7 camera = PiCamera()
8 camera.resolution = (640, 480)
9 camera.framerate = 32
10 rawCapture = PiRGBArray(camera, size=(640, 480))
11 # Set input files
12 cascPath = "haarcascade_frontalface_default.xml"
13
14 # Create the haar cascade
15 faceCascade = cv2.CascadeClassifier(cascPath)
16
17 # allow the camera to warmup
18 time.sleep(0.1)
19
20 # capture frames from the camera
21 for frame in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
22     # grab the raw NumPy array representing the image, then initialize the timestamp
23     # and occupied/unoccupied text
24     image = frame.array
25     # Convert BGR to HSV
26     hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
27     # define range of white color in HSV
28     lower = np.array([0, 0, 100])
29     upper = np.array([35, 55, 255])
30     # Threshold the HSV image to get only the desired color
31     mask = cv2.inRange(hsv, lower, upper)
32
33     # Use morphology to remove noise
34     kernel = np.ones((11, 11), np.uint8)
35     morpho = cv2.morphologyEx(mask, cv2.MORPH_OPEN, kernel)
36     morpho = cv2.morphologyEx(morpho, cv2.MORPH_CLOSE, kernel)
37
38     # Find and draw contours
39     im2, contours, hierarchy = cv2.findContours(morpho, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
40     draw = image.copy()
41     cv2.drawContours(draw, contours, -1, (0, 255, 0), 2)
42
43     # Iterate over all contours and draw enclosing circle
44     for c in contours:
45         (x, y), radius = cv2.minEnclosingCircle(c)
46         center = (int(x), int(y))
47         radius = int(radius)
48         cv2.circle(image, center, radius, (0, 0, 255), 2)
49
50     # show the frame
51     cv2.imshow("Frame", image)
52     key = cv2.waitKey(1) & 0xFF
53
54     # clear the stream in preparation for the next frame
55     rawCapture.truncate(0)
56
57     # if the `q` key was pressed, break from the loop
58     if key == ord("q"):
59         break

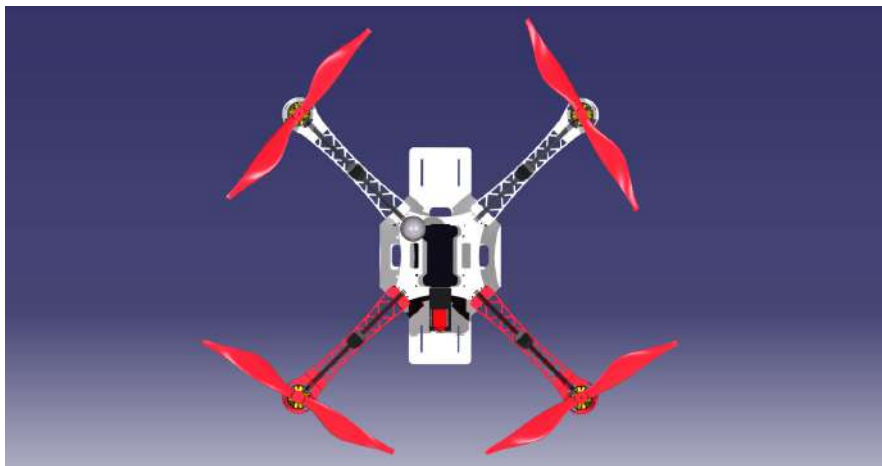
```

Figure A.2: Code for detection from live camera stream.

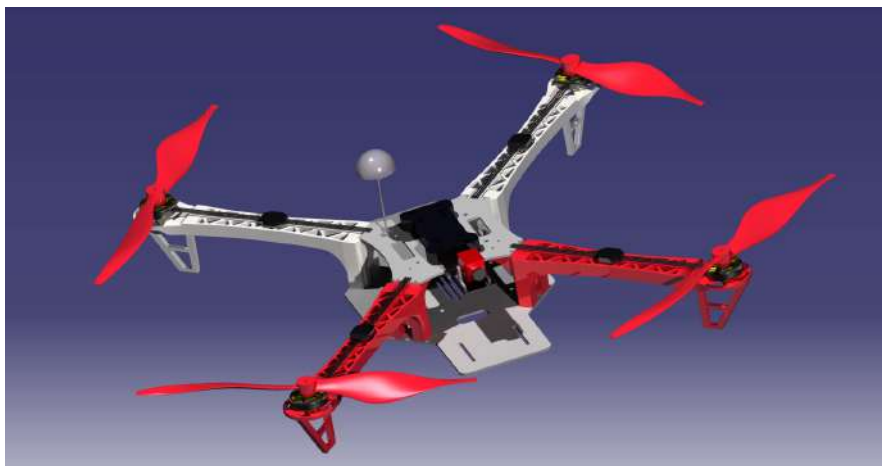
Figure A.3: Three views of FSD CAD-model.



(a) Front view of FSD CAD-model.



(b) Top view of FSD CAD-model



(c) Cinematic view of FSD CAD-model.